HIGH SPEED TEXT SEARCH SYSTEM

HSTS SOFTWARE
LISTINGS

VOL. 1 OF 5

Master
Computer
Support

STAT

NGA review(s) completed.

HSTS MASTER COMPUTER SOFTWARE LISTINGS

SL120100

VOLUME 1 of 5

Prepared for:

Central Intelligence Agency
Washington, DC 20505

[redacted] R80-016 STAT

March 1980

STAT

[redacted]

Approved For Release 2005/07/28 : CIA-RDP85-00514R000100030001-3

TABLE OF CONTENTS

10- 2 MACRO'S AND CONSTANTS
11- 23 ASSEMBLY-TIME DATA DEFINITION
12- 83 INITIALIZE MASTER COMMON

```
                                .TITLE  INIT
                                .SBTTL  MACRO'S AND CONSTANTS
                        ;
                                .MCALL  QIOW$C,ALUN$S
                                .MCALL  NMBLK$,FDOF$L,FCSBT$,FINIT$
                                .MCALL  FDBDF$,FDRC$A,FDBK$A,FDOP$A,FSRSZ$
                                .MCALL  RQST$S,EXIT$S,RQST$C,RUN$C
                        ;
                                .GLOBL  DIRP74,DIRP75,DIRP31,DIRP32
                                .GLOBL  .GTDIR
                                .GLOBL  PUTSSQ,GETFRE
                                .GLOBL  SUST,SUINDX
                                .GLOBL  STATSS,STATSE
                        ;
                        ;LUNS
000001                  DPLUN=1
000002                  XMLUN=2
                        ;
                        ;MISC EQUATES
000001                  EF.IO=1         ;I/O EVENT FLAG
                        ;
```

```
                                        .SBTTL  ASSEMBLY-TIME DATA DEFINITION
                                ;
                                ;DUMMY FDB
                                        FDOF$L
                                        FCSBT$
                                ;
000000                          FDB:    FDBDF$
000140                                  FDRC$A  FD.RWM
000140                                  FDBK$A  BUFFER,N,BUFB,,EF,IO,IOSTAT
000140                                  FDOP$A  DPLUN
                                ;
000140                                  FSRSZ$  0
                                ;
                                ;DUMMY NAME BLOCK TO GET DIRECTORY FID'S
000140                          DUMNBK: NMBLK$  ,,,SY,0
                                ;
                                ;DIRECTORY NAME DESCRIPTORS
000176  000005                 DIRDS1: .WORD   5
000200  000202'                         .WORD   DIRDT1
000202      133     067     054 DIRDT1: .ASCII  /[7,4]/
    000205      064     135
                                        .EVEN
000210  000005                 DIRDS2: .WORD   5
000212  000214'                         .WORD   DIRDT2
000214      133     067     054 DIRDT2: .ASCII  /[7,5]/
    000217      065     135
                                        .EVEN
000222  000007                 DIRDS3: .WORD   7
000224  000226'                         .WORD   DIRDT3
000226      133     063     060 DIRDT3: .ASCII  /[300,1]/
    000231      060     054     061
    000234      135
                                        .EVEN
000236  000007                 DIRDS4: .WORD   7
000240  000242'                         .WORD   DIRDT4
000242      133     063     060 DIRDT4: .ASCII  /[300,2]/
    000245      060     054     062
    000250      135
                                        .EVEN
                                ;
                                ;DIRECTORY ADDRESS TABLE
000252  000176'                 DIRTBL: .WORD   DIRDS1
000254  000000G                         .WORD   DIRP74
000256  000210'                         .WORD   DIRDS2
000260  000000G                         .WORD   DIRP75
000262  000222'                         .WORD   DIRDS3
000264  000000G                         .WORD   DIRP31
000266  000236'                         .WORD   DIRDS4
000270  000000G                         .WORD   DIRP32
                                ;
                                ;MISC LOCATIONS
000272                          BUFFER:                 ;DUMMY BUFFER
000272                          IOSTAT: .BLKW   2       ;I/O STATUS BLOCK
                                ;
                                ;DMCIN TASK NAME TABLE
                                ;
000276                          DMCTBL:
```

```
74          000000                  N=0
75          000002·                 .REPT·   N,SUNT·
76                                  .IRP·    Z,<\N>
77                                           .RAD50   /DMCIN'Z'/·
78                                  .ENDR·
79                                  N=N+1
80                                  .ENDR·
81                                  ;
```

```
000000                  N=0
000002                  .REPT   N,SUNT
                        Z,<\N>
                        .RAD50  /DMCIN:Z'/
                        .ENDR
                        N=N+1
                        .ENDR
                        ;
```

```
                                        .SBTTL  INITIALIZE MASTER COMMON
                                ;
                                INIT:
                                        FINIT$
                                ;
                                ;INIT FID'S OF SYSTEM DIRECTORIES
000312  012700  000000'                MOV     #FDB,R0                 ;FDB ADDRESS
000316  012701  000140'                MOV     #DUMNBK,R1              ;DUMMY NAME BLOCK
                                ;
000322  012703  000252'                MOV     #DIRTBL,R3              ;DIRECTORY ADDRESS TABLE
000326  012704  000004                 MOV     #4,R4                   ;LOOP COUNT
                                ;
000332  012302                 DIRLOP: MOV     (R3)+,R2                ;DIRECTORY DESCRITOR ADDRESS
000334                                 CALL    .GTDIR                  ;GET DIRECTORY FID
000340  012302                         MOV     (R3)+,R2                ;COMMON MEMORY ADDRESS
000342  016712  177616                 MOV     DUMNBK+N.DID,(R2)       ;COPY FID TO MCOM
000346  016762  177614  000002         MOV     DUMNBK+N.DID+2,2(R2)
000354  016762  177610  000004         MOV     DUMNBK+N.DID+4,4(R2)
000362  077415                         SOB     R4,DIRLOP
                                ;
```

```
                                        ;
                                        ;
                                        ; INIT SUST
000364  012767  177777  177776G         MOV     #-1,SUST-2
000372  012703  000001         ISULOP:  MOV     #N,SUNT-1,R0            ;SET SU'S TO IDLE
000376  006300                 1$:      ASL     R0
000400  016001  000000G                  MOV     SUINDX(R0),R1
000404  006200                          ASR     R0
000406  012761  000000  000000          MOV     #SU.IDL,SS,STT(R1)
000414  005300                          DEC     R0
000416  100367                          BPL     1$
                                        ;
                                        ;
                                        ; INIT HOURLY STATS
                                        ;
000420  012700  000000G        INHSTT:  MOV     #STATSS,R0              ;CLEAR STATS AREA
000424  012701  000000C                  MOV     #<STATSE-STATSS>/2,R1
000430  005020                 1$:      CLR     (R0)+
000432  077102                          SOB     R1,1$
                                        ;
000434                                  RUN$C   HRSTAT,,,,1,1,45,,2     ;SCHEDULE HOURLY STATS
                                        ;
                                        ;
                                        ;INIT THE DMC TO THE SEARCH UNITS
000442  012700  000001         IDMCLP:  MOV     #N,SUNT-1,R0            ;HIGH SU #
000446  012701  000302'                 MOV     #DMCTBL+<4*<N,SUNT-1>>,R1 ;DMCIN TASK NAME
000452                          L1:      ALUN$S  #XMLUN,#"XM,R0
                                        ;
000472                                  QIOW$C  IO,TRM,XMLUN,EF,IO
000500                                  QIOW$C  IO,INL,XMLUN,EF,IO
000506                                  ROST$S  R1                      ;RUN DMCIN
000532  162701  000004                  SUB     #4,R1
000536  005300                          DEC     R0                      ;LOOP FOR ALL SU
000540  100344                          BPL     L1
                                        ;
                                        ;
                                        ; START THE SCHEDULER
                                        ;
000542                                  ROST$C  SCHD00                  ;BATCH 0 IS OFF
                                        ;
                                        ; PRIME SCHD00
000550                                  CALL    GETPRE                  ;FORCE SCHEDULER INITIALIZATION
000554  112762  000000  000002          MOVB    #XMSCHED,2(R2)
000562  112762  000000  000003          MOVB    #0,3(R2)
000570  005062  000004                  CLR     4(R2)
000574                                  CALL    PUTSSQ
                                        ;
```

```
152 000600                          EXIT$S
153
154         000306'                 .END    INIT
```

```
BITVAL=  000000          B.QSPL  000316          010 FA.SEQ=  040000          FO.APD=  000106          IOSTAT  000272R
BIT0  =  000001          B.QTTM  000076          010 FA.SHR=  000040          FO.MFY=  000002          IO.INL=  ****** GX
BIT1  =  000002          B.QUQP  000056          010 FA.TMP=  000020          FO.RD =  000001          IO.TRM=  ****** GX
BIT10 =  002000          B.SFDB  000010          010 FA.WCK=  020000          FO.UPD=  000006          ISULOP  000372R
BIT11 =  004000          B.SIZE  000772          010 FA.WRT=  000002          FO.WRT=  000016          L1      000452R
BIT12 =  010000          B.SNDP  000012          010 FDB     000000R          F.ACTL=  000076          M       =  000002
BIT13 =  020000          B.SSQ   000004          010 FD.BLK=  000010          F.ALOC=  000040          N       =  000002
BIT14 =  040000          B.SYQF  000050          010 FD.CCL=  000002          F.BBFS=  000062          NB.DEV=  000200
BIT15 =  100000          B.SYAT  000044          010 FD.COM=  020000          F.BDB =  000070          NB.DIR=  000100
BIT2  =  000004          B.SYTB  000053          010 FD.CR =  000002          F.BGBC=  000057          NB.NAM=  000004
BIT3  =  000010          B.UUOC  000110          010 FD.DIR=  000010          F.BLDN=  000026          NB.SDI=  000400
BIT4  =  000020          CF.10 =  000070              FD.FID  000000      003 F.BKDS=  000020          NB.SD2=  001000
BIT5  =  000040          CF.U2 =  000067              FD.FNB  000006      003 F.BKEF=  000050          NB.SNM=  000040
BIT6  =  000100          CF.B4 =  000066              FD.FTH=  000001          F.BKP1=  000051          NB.STP=  000020
BIT7  =  000200          CF.B6 =  000065              FD.FVR  000004      003 F.BKST=  000024          NB.SVR=  000010
BIT8  =  000400          CF.DR0=  000064              FD.F11=  040000          F.BKVB=  000064          NB.TYP=  000002
BIT9  =  001000          CF.DR1=  000063              FD.INS=  000010          F.CHR =  000075          NB.VER=  000001
BS.CLS=  000002          CH.AND=  000001              FD.ISP=  002000          F.CNTG=  000034          N.BFAC=  000004
BS.DBU=  000004          DBSLEN=  000116              FD.LEN  000010      003 F.DFNB=  000046          N.BUSH=  000006
BS.INA=  000000          DH.BF0  000002          005 FD.MNT=  100000          F.DSPT=  000044          N.BUSY=  000001
BS.OPN=  000001          DH.BF1  000004          005 FD.USP=  004000          F.DVNM=  000134          N.BUFH=  000000
BS.SRC=  000003          DH.CTL  000000          005 FD.PLC=  000004          F.EFNK=  000010          N.BUFW=  000000
BUFFER  000272R          DH.DMC  000010          005 FD.PRN=  000004          F.EFN =  000050          N.DID =  006824
BYTE0 =  000000          DH.FLG  000006          005 FD.PSE=  010000          F.EOBB=  000032          N.DVNM=  000032
BYTE1 =  000001          DIRDS1  000176R              FD.RAH=  000001          F.ERR =  000052          N.FID =  000000
BYTE2 =  000002          DIRDS2  000210R              FD.RAN=  000002          F.FACC=  000045          N.FNAM=  000006
BYTE3 =  000003          DIRDS3  000222R              FD.REC=  000001          F.FFBY=  000014          N.FOS =  000764
BYTE4 =  000004          DIRDS4  000236R              FD.RWM=  000001          F.FNAM=  000110          N.FTYP=  000014
BYTE5 =  000005          DIRDT1  000202R              FD.SDI=  000020          F.FNB =  000102          N.FVER=  000006
BYTE6 =  000006          DIRDT2  000214R              FD.SQD=  000040          F.FTYP=  000116          N.NEXT=  000022
BYTE7 =  000007          DIRDT3  000226R              FD.TTY=  000004          F.FVER=  000120          N.PKSZ=  000020
BYTE8 =  000010          DIRDT4  000242R              FD.UBH=  000002          F.HIBK=  000004          N.PKTS=  000043
BYTE9 =  000011          DIRLOP  000332R              FF.CHR=  000005          F.LUN =  000042          N.QURY=  000031
BYTVAL=  000012          DIRP31=  ****** G            FF.NV =  000003          F.MBCT=  000054          N.STAT=  000020
B.BSTA  000054      010  DIRP32=  ****** G            FF.POE=  000002          F.MBC1=  000055          N.SUNT=  000002
B.CNTX  000046      010  DIRP74=  ****** G            FF.RWD=  000001          F.MBFG=  000056          N.UNIT=  000034
B.CQUQ  000060      010  DIRP75=  ****** G            FF.RWF=  000006          F.NRBD=  000024          PUTSSQ=  ****** G
B.FEMA  000132      010  DIRTBL  000252R              FF.SPC=  000004          F.NREC=  000030          QE.ROI=  000144
B.FEMB  000142      010  DNSTBL  000276R              FN.DBR  000026      011 F.OVBS=  000030          Q.FDSC  000004      007
B.FEMC  000152      010  DN.DCK  000000          013 FN.DBS  000022      011 F.RNCC=  000016          Q.IOAE=  000012
B.FFSA  000202      010  DN.NTP  000004          013 FN.DHR  000040      011 F.RNTT=  000001          Q.IOEF=  000006
B.FFSB  000212      010  DN.NXT  000006          013 FN.EMA  000012      011 F.RCNM=  000034          Q.IOFN=  000002
B.FFSC  000222      010  DN.ROT  000002          013 FN.EMB  000014      011 F.RCTL=  000017          Q.IOLU=  000004
B.FMHR  000172      010  DN.SIZ  000010          013 FN.EMC  000016      011 F.RSIZ=  000002          Q.IOPL=  000014
B.FOLS  000162      010  DFLUN =  000001              FN.FSA  000000      011 F.RTYP=  000000          Q.IOPR=  000007
B.FSAZ  000100      010  DUMMBK  000140R              FN.FSB  000002      011 F.SEQN=  000100          Q.IOSW=  000010
B.FSIZ  000102      010  EF.IO =  000001              FN.FSC  000004      011 F.SPDV=  000072          Q.NOBY  000000      007
B.FSCZ  000104      010  FA.APD=  000100              FN.LGO  000034      011 F.SPUN=  000074          Q.NUHL  000002      007
B.HBLK  000120      010  FA.CKC=  000010              FN.LGU  000036      011 F.STBK=  000036          Q.SIZE  000004      007
B.HDOC  000114      010  FA.DLK=  001000              FN.MFO  000024      011 F.UNIT=  000136          R.FIX =  000001
B.HRLP  000126      010  FA.ENB=  100000              FN.MHR  000010      011 F.URBD=  000020          R.OSFC  000014
B.HRLR  000122      010  FA.EXC=  002000              FN.NMB  000041      011 F.VDN =  000064          R.OSPC  000013
B.HRLU  000124      010  FA.EXT=  000004              FN.OLS  000006      011 F.VBSZ=  000060          R.OSPN  000015
B.NMBR  000052      010  FA.NSP=  000100              FN.QRY  000020      011 GETFRE=  ****** G        R.OSPR  000012
B.NQRY  000232      010  FA.POS=  010000              FN.SF0  000030      011 IDMCLP  000442R          R.OSTH  000002
B.OLSZ  000106      010  FA.RD =  000001              FN.SF1  000032      011 INHSTT  000420R          R.SEQ.  000003
B.UMAP  000234      010  FA.RUD=  004000              FN.SHD  000042      011 INIT    000306R          R.UNGC=  000015
```

```
R.UNPC=  000014          SR.NDS  000036      002 ST.LEN  000044      006 S.HRL =  000240          XFDSMR=  000007
R.UNPH=  000006          SR.NIN  000050      002 ST.QRY  000002      006 S.NFEN=  000020          XGTSPE=  000014
R.UNPR=  000012          SR.NIP  000022      002 ST.QSZ  000034      006 UN.NTP  000004      012 XHITSK=  000011
R.UNRM=  000022          SR.SDB  000032      002 ST.SCH  000040      006 UN.NXT  000006      012 XHLMEM=  000002
R.UNRU=  000024          SR.SRC  000002      002 ST.UHL  000004      006 UN.ROT  000002      012 XHOTSK=  000010
R.UNSM=  000016          SR.SUN  000000      002 ST.XLT  000014      006 UN.SIZ  000010      012 XMLUN=  000002
R.UNSU=  000020          SR.TUS  000056      002 SUINDX= ******  G       UN.SRC  000000      012 XMSCNE=  000005
R.UNTM=  000002          SR.USL  000052      002 SUST  = ******  G       UN.TYP  000001      012 XOTS  =  000003
R.VAR =  000002          SR.VR   000004      002 SU.DBU= 000004          WORD0 =  000000          XOT0  =  000001
SR.ARE   000114      002 SR.WIN  000024      002 SU.DON= 000006          WORD1 =  000002          XSULOA=  00A005
SR.ARS   000106      002 SR.WIP  000016      002 SU.IDL= 000000          WORD2 =  000004          $$$   =  000196R     015
SR.DAY   000010      002 SS.FID  000002      004 SU.LOD= 000001          WORD3 =  000006          $$VARS=  000000
SR.DLT   000014      002 SS.FNB  000010      004 SU.SRC= 000002          WORD4 =  000010          $$$DSI=  000016
SR.ECB   000047      002 SS.FVR  000006      004 SU.SRR= 000005          WORD5 =  000012          $$XT1 =  000000
SR.ECM   000046      002 SS.LEN  000012      004 SU.XPD= 000003          WORD6 =  000014          $$XT2 =  000001
SR.ECL   000059      002 SS.TT   000000      003 S.DFHD= 000020          WORD7 =  000016          .FINI=  000000 G
SR.FIB   000012      002 STATSE= ******  G       S.FATT= 000015          WORD8 =  000020          .FSRCB=  000000 G
SR.GRE   000100      002 STATUS= ******  G       S.FDB = 000140          WORD9 =  000022          .GIDR=  000000 G
SR.GRS   000072      002 ST.NSZ  000020      006 S.FNAM= 000006          WRDVEL=  000024          ...ERL=  000000
SR.LEN   000122      002 ST.RSZ  000024      006 S.FNB = 000036          XBATCH=  000013          ...PC1=  000000
SR.LIN   000066      002 ST.BTC  000000      006 S.FNBW= 000017          XDTADA=  000004          ...PC2=  000000
SR.LIP   000062      002 ST.CSZ  000030      006 S.FNTY= 000004          XDUMP0=  000012          ...PC3=  000000
SR.NOF   000006      002 ST.HRL  000010      006 S.FTYP= 000002          XPRIM=  000006           ...TPC=  000000
SR.NDC   000042      002

.ABS.    000000     000
         000606     001
SRCOFF   000122     002
FDSCOF   000010     003
SUSOFF   000012     004
DHROFF   000012     005
STTOFF   000044     006
OSPLOF   000014     007
BSTOFF   000772     010
FNOFFS   000044     011
UNODOF   000010     012
DNODOF   000010     013
$$FSR1   000000     014
$DPB$$   000124     015
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  6142 WORDS  (24 PAGES)
DYNAMIC MEMORY:  7028 WORDS  ( 27 PAGES)
ELAPSED TIME:  00:00:40
[INIT,INIT/-SP=C20,1JP,M,INIT
```

10- 2 SCHEDULER ROOT SEGMENT

```
                        .TITLE  MSCHED
                        .SBTTL  SCHEDULER ROOT SEGMENT
                        ;
                        ;
                        .MCALL  FDBDF$,FDRC$A,FDBK$A,FDOP$A,FSRSZ$
                        .MCALL  FINIT$,FDOF$L,FCSBT$
                        .MCALL  ALTP$S
                        ;
                        ;
                        .GLOBL  BSTPTR,GETFRE,PUTSSQ
                        .GLOBL  STTENT
                        .GLOBL  BN              ;BATCH # - GLBDEF'D IN TKB
                        ;
                        ;
                        ; SCHEDULER'S FDB
                        ;
000000                  FDOF$L
000000                  FCSBT$
                        ;
000000                  FDB:    FDBDF$
000140                  FDRC$A  FD,RWM
000140                  FDBK$A  DATBUF,N.BUFB,,1,IOSTAT
000140                  FDOP$A  1
                        ;
000140                  FSRSZ$  0
                        ;
                        ;
000140                  IOSTAT: .BLKW   2       ;I/O STATUS BLOCK
000144                  DATBUF:                 ;REENTRANT SAVE AREA FOR STT
000144                  .BLKW   <N.QURY*2>+4    ;BIG ENOUGH TO HOLD BATCH CUT-OFF MSG
                        ;
```

```
 33 000320                              .MSCHED:.
 34 000320                                      FINIT$
 35                                     ;
 36 000324  012705  000000G.                    MOV.    #BN,R5                  ;GET.OUR.BST.ADDRESS
 37 000330  016505  000000G.                    MOV.    BSTPTR(R5),R5
 38                                     ;
 39 000334  112765  000001  000053              MOVB.   #BS.OPN,B.STTE(R5)       ;BATCH.IS.OPEN.
 40 000342. 012765  000000' 000010              MOV.    #FDB.B.SFDB(R5)          ;SCHEDULER'S.FDB.
 41 000350  005065  000124                      CLR.    B.HRLW(R5)               ;INIT.HRL.WORDS.
 42.000354  012765  000240  000120              MOV.    #S.HRL,B.HBLK(R5)
 43                                     ;
 44 000362.                                     CALL.   GETFRE.                  ;PRIME.SSQ TO.LOOK.-
 45 000366  112762. 000000  000002.             MOVB.   #XMSCHED,2(R2)           ;  AT.QUQ.
 46 000374  112762. 000004  000003              MOVB.   #4.3(R2)
 47 000402.                                     CALL.   PUTSSQ.
 48                                     ;
 49 000406                                      ALTP$S. ,#40.                    ;RUNNING.PRIORITY.
 50                                     ;
 51 000424  000167  000000G.                    JMP.    STTENT.                  ;START.STATE.TABLE.
 52.                                    ;
 53                                     ;
 54         000320'                             .END.   MSCHED.
```

```
BITVAL=.000000          B.QMAP  000234          010 FD.INS=.000010          F.CHR =.000075          N.DID =.000024
BIT0  =.000001          B.QSPL  000316          010 FD.ISP=.002000          F.CNTG=.000034          N.DVNM=.000032
BIT1  =.000002          B.QTTM  000076          010 FD.LEN.000010       003 F.DFNB=.000046          N.FID =.000000
BIT10 =.002000          B.QUOP  000056          010 FD.MNT=.100000          F.DSPT=.000044          N.FNAM=.000006
BIT11 =.004000          B.SFDB  000010          010 FD.OSP=.004000          F.DVNM=.000134          N.FOS =.000764
BIT12 =.010000          B.SIZE  000772          010 FD.PLC=.000004          F.EFBK=.000010          N.FTYP=.000014
BIT13 =.020000          B.SNDP  000012          010 FD.PRN=.000004          F.EFN =.000050          N.FVER=.000016
BIT14 =.040000          B.SSQ.  000004          010 FD.PSE=.010000          F.EOBB=.000032          N.NEXT=.000022
BIT15 =.100000          B.SSQF  000050          010 FD.RAH=.000001          F.ERR =.000052          N.PKSZ=.000020
BIT2  =.000004          B.STAT  000044          010 FD.RAN=.000002          F.FACC=.000043          N.PKTS=.000043
BIT3  =.000010          B.STTE  000053          010 FD.REC=.000001          F.FFBY=.000014          N.QURY=.000031
BIT4  =.000020          B.UDOC  000110          010 FD.RWM=.000001          F.FNAM=.000110          N.STAT=.000020
BIT5  =.000040          CF.B0 =.000070              FD.SDI=.000020          F.FNB =.000102          N.SUNT=.000002
BIT6  =.000100          CF.B2 =.000067              FD.SQD=.000040          F.FTYP=.000116          N.UNIT=.000034
BIT7  =.000200          CF.B4 =.000066              FD.TTY=.000004          F.FVER=.000120          PUTSSQ=****** G
BIT8  =.000400          CF.B6 =.000065              FD.WBH=.000002          F.HIBK=.000004          QE.RO1=.000144
BIT9  =.001000          CF.DR0=.000064              FF.CHR=.000005          F.LUN =.000042      007 Q.FDSC.000004      007
BN    =****** G         CF.DR1=.000063              FF.NV =.000003          F.MBCT=.000054      007 Q.NOBK.000000      007
BSTPTR=****** G         CH.AND=.000001              FF.POE=.000002          F.MBC1=.000055      007 Q.NUHL.000002      007
BS.CLS=.000002          DATBUF  000144R             FF.RWD=.000001          F.MBFG=.000056      007 Q.SIZE.000014      007
BS.DBU=.000004          DBSLEN=.000116              FF.RWF=.000006          F.NRBD=.000024          R.FIX =.000001
BS.INA=.000000          DH.BF0  000002          005 FF.SPC=.000004          F.NREC=.000030          R.SEQ =.000003
BS.OPN=.000001          DH.BF1  000004          005 FN.DBR.000026       011 F.OVBS=.000030          R.VAR =.000002
BS.SRC=.000003          DH.CTL  000000          005 FN.DBS.000022       011 F.RACC=.000016      002 SR.ARE.000114      002
BYTE0 =.000000          DH.DMC  000010          005 FN.DHR.000040       011 F.RATT=.000001      002 SR.ARS.000106      002
BYTE1 =.000001          DH.FLG  000006          005 FN.EMA.000012       011 F.RCNM=.000034      002 SR.DAY.000010      002
BYTE2 =.000002          DN.DCK  000000          013 FN.EMB.000014       011 F.RCTL=.000017      002 SR.DLT.000014      002
BYTE3 =.000003          DN.NTP  000004          013 FN.EMC.000016       011 F.RSIZ=.000002      002 SR.ECB.000047      002
BYTE4 =.000004          DN.NXT  000006          013 FN.FSA.000000       011 F.RTYP=.000000      002 SR.ECH.000046      002
BYTE5 =.000005          DN.ROT  000002          013 FN.FSB.000002       011 F.SEQN=.000100      002 SR.ECL.000050      002
BYTE6 =.000006          DN.SIZ  000010          013 FN.FSC.000004       011 F.SPDV=.000072      002 SR.FIB.000012      002
BYTE7 =.000007          FA.APD=.000100              FN.LGQ.000034       011 F.SPUN=.000074      002 SR.GRE.000100      002
BYTE8 =.000010          FA.CRE=.000010              FN.LGU.000036       011 F.STBK=.000036      002 SR.GRS.000072      002
BYTE9 =.000011          FA.DLK=.001000              FN.MFO.000024       011 F.UNIT=.000136      002 SR.LEN.000122      002
BYTVAL=.000012          FA.ENB=.100000              FN.MHR.000010       011 F.URBD=.000020      002 SR.LTN.000066      002
B.BSTA.000054       010 FA.EXC=.002000              FN.NMB.000044       011 F.VBN =.000064      002 SR.LIP.000062      002
B.CNTX.000046       010 FA.EXT=.000004              FN.QLS.000006       011 F.VBSZ=.000060      002 SR.MON.000006      002
B.COUQ.000060       010 FA.NSP=.000100              FN.QRY.000020       011 GETFRE=****** G     002 SR.NDC.000042      002
B.FEMA.000132       010 FA.POS=.010000              FN.SF0 000030       011 IOSTAT.000140R      002 SR.NDS.000036      002
B.FEMB.000142       010 FA.RD =.000001              FN.SF1 000032       011 M.    =.000062      002 SR.NIN.000030      002
B.FEMC.000152       010 FA.RWD=.004000              FN.SHD.000042       011 MSCHED.000320R      002 SR.NIP.000022      002
B.FFSA.000202       010 FA.SEQ=.040000              FO.APD=.000106          N.    =.000002      002 SR.SDB.000032      002
B.FFSB.000212       010 FA.SHR=.000040              FO.MFY=.000002          NB.DEV=.000200      002 SR.SRC.000002      002
B.FFSC.000222       010 FA.TMP=.000020              FO.RD =.000001          NB.DIR=.000100      002 SR.SUN.000000      002
B.FMHR.000172       010 FA.WCK=.020000              FO.UPD=.000006          NB.NAM=.000004      002 SR.TWS.000056      002
B.FQLS.000162       010 FA.WRT=.000002              FO.WRT=.000016          NB.SD1=.000400      002 SR.WSL.000052      002
B.FSAZ.000100       010 FDB.    000000R             F.ACTL=.000076          NB.SD2=.001000      002 SR.YR .000004      002
B.FSBZ.000102       010 FD.BLK=.000010              F.ALOC=.000040          NB.SNM=.000040      002 SR.1IN.000024      002
B.FSCZ.000104       010 FD.CCL=.000002              F.BBFS=.000062          NB.STP=.000020      002 SR.1IP.000016      002
B.HBLK.000120       010 FD.COM=.020000              F.BDB =.000070          NB.SVR=.000010      002 SS.FID.000002      004
B.HDOC.000114       010 FD.CR =.000002              F.BGBC=.000057          NB.TYP=.000002      002 SS.FNB.000010      004
B.HRLP.000126       010 FD.DIR=.000010              F.BKDN=.000026          NB.VER=.000001      002 SS.FVR.000006      004
B.HRLR.000122       010 FD.FID  000000          003 F.BKDS=.000020          N.BFAC=.000004      002 SS.LEN.000012      004
B.HRLU.000124       010 FD.FNB  000006          003 F.BKEF=.000050          N.BHGH=.000006      002 SS.STT.000000      004
B.NMBR.000052       010 FD.FTN=.000001              F.BKP1=.000051          N.BTCH=.000004      002 STTENT=****** G
B.NQRY.000232       010 FD.FVR  000004          003 F.BKST=.000024          N.BUFB=.004000      002 ST.ASZ.000020      006
B.QLSZ.000106       010 FD.Fi1=.040000              F.BKVB=.000064          N.BUFW=.002000      002 ST.BSZ.000024      006
```

```
ST.BTC  000000      006 SU.SRC= 000002      WN.NTP  000004      012 WORD6 = 000014      XHLMER= 000002
ST.CSZ  000030      006 SU.SRR= 000005      WN.NXT  000006      012 WORD7 = 000016      XHOTSK= 000010
ST.HRL  000010      006 SU.XPD= 000003      WN.ROT  000002      012 WORD8 = 000020      XMSCHE= 000000
ST.LEN  000044      006 S.AFHD= 000020      WN.SIZ  000010      012 WORD9 = 000022      XQTS  = 000003
ST.ORY  000002      006 S.FATT= 000016      WN.SRC  000000      012 WRDVAL= 000024      XQT0  = 000001
ST.OSZ  000034      006 S.FDB = 000140      WN.TYP  000001      012 XBATCH= 000013      XSULOA= 000005
ST.SCH  000040      006 S.FNAM= 000006      WORD0 = 000000          XDBLNA= 000004      .FINIT= ****** G
ST.UHL  000004      006 S.FNB = 000036      WORD1 = 000002          XDBPRO= 000012      .FSRLB= ****** G
ST.XLT  000014      006 S.FNBW= 000017      WORD2 = 000004          XDMCIN= 000006      ...GBL= 000000
SU.DBU= 000004          S.FNTY= 000004      WORD3 = 000006          XFOSMR= 000007      ...PC1= 000000R
SU.DON= 000006          S.FTYP= 000002      WORD4 = 000010          XGTSKE= 000A14      ...PC2= 000140R
SU.IDL= 000000          S.HRL = 000240      WORD5 = 000012          XHITSK= 000011      ...TPC= 000020

. ABS.  000000      000
        000430      001
SRCOFF  000122      002
FDSCOF  000010      003
SUSOFF  000012      004
DHROFF  000012      005
STTOFF  000044      006
QSPLOF  000014      007
BSTOFF  000772      010
FNOFFS  000044      011
WNODOF  000010      012
DNODOF  000010      013
$$FSR1  000000      014
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  4938 WORDS  (20 PAGES)
DYNAMIC MEMORY:  5972 WORDS  ( 22 PAGES)
ELAPSED TIME:  00:00:27
MSCHED,MSCHED/-SP=[20,1]P,M,MSCHED
```

10- 2 STATE TRANSITION TABLE PROCESSING
11- 35 NEXT STATE TRANSITION ROUTINE
12- 179 STATE TABLE MACROS
14- 2 STATE TRANSITION TABLES
15- 223 MDBU STATES
16- 254 BATCH FAILURE PATHS
18- 2 DECISION ROUTINES
20- 2 CONDITION MATCH ROUTINES
21- 411 MDBU CONDITION MATCH ROUTINES
22- 451 ERROR PATH CONDITION ROUTINES
24- 2 STATE TABLE SUPPORT ROUTINES

```
                                .TITLE  STT
                                .SBTTL  STATE TRANSITION TABLE PROCESSING
                               ;
                               ;
                                .MCALL  CLEF$S,WTSE$S
                                .MCALL  CMKT$S,MRKT$S,EXIT$S
                                .MCALL  SETF$S,ASTX$S
                                .MCALL  SDAT$S,RSUM$S,ROST$S
                                .MCALL  ALTP$S,GTIM$S,QIOW$S
                                .MCALL  FDAT$R,OFNB$W,WRITE$,WAIT$,CLOSE$
                                .MCALL  FDOF$L,FCSBT$
                               ;
                                .GLOBL  PUTFRE,PUTOUT,SEZLOK,RLSLOK
                                .GLOBL  GETFRE,GETOUQ,GETSSQ,PUTSSQ
                                .GLOBL  BSTPTR,SUINDX,SUST
                                .GLOBL  BLDNFL,BLDEFL,.DLFNB
                                .GLOBL  SYSFLG
                                .GLOBL  CHSTAT,$DDIV,SRECPT
                               ;
                               ;
000000                          FDOF$L
000000                          FCSBT$
                               ;
                               ;
                               ;
                               ; EQUATES
                               ;
        177777                 DN=-1            ;DECISION NODE INDICATOR
        000000                 WN=0             ;WAIT NODE INDICATOR
        000004                 TRCLUN=4         ;TRACE OUTPUT
        000006                 COLUN=6          ;CONSOLE OUTPUT
                               ;
                               ;
```

```
                                        .SBTTL  NEXT STATE TRANSITION ROUTINE
                                ;
                                ; STATE TABLE TRANSITION ROUTINE
                                ;
                                ; ENTRY FROM SCHEDULER ROOT TASK - R5= BST ADDRESS
                                ;
000000  012765  000452' 000046  STTENT::MOV     #TOPNOD,B.CNTX(R5)      ;START AT TOP NODE
                                ;
                                ;
                                ; GET NEXT STATE
000006                          NEXTST:
000006  005775  000046                  TST     @B.CNTX(R5)             ;BRANCH IF NO
000012  001412                          BEQ     NOTIME                  ;  TIME-OUT REQUIRED
000014                                  MRKT$S  ,@B.CNTX(R5),#2,#TIMAST ;TIME OUT
                                ;
000040  005765  000002          NOTIME: TST     B.SSQ-2(R5)             ;BRANCH IF HAVE SSQ ENTRY
000044  001042                          BNE     DATSSQ
                                ;
000046  116500  000050          WAITSQ: MOVB    B.SSQF(R5),R0           ;WAIT FOR NEW SSQ ENTRY OR
000052                                  WTSE$S  R0                      ;  TIME OUT
000062                                  CLEF$S  R0
000072  032765  100000  000044          BIT     #BIT15,B.STAT(R5)       ;BRANCH IF NOT T/O
000100  001757                          BEQ     NOTIME
                                ;
                                ; GOT WAIT NODE TIMEOUT
000102  016504  000046                  MOV     B.CNTX(R5),R4           ;STATE TABLE START FOR NODE
000106  062704  000002                  ADD     #2,R4
000112  005764  000000          1$:     TST     WN.SRC(R4)              ;FIND T/O ENTRY
000116  100403                          BMI     2$
000120  062704  000010                  ADD     #WN.SIZ,R4
000124  000772                          BR      1$
                                ;TIMEOUT ENTRY
000126  016465  000006  000046  2$:     MOV     WN.NXT(R4),B.CNTX(R5)   ;NEXT NODE
                                ;
                                ; CALL TIMEOUT ROUTINE (NO PACKET)
                                ;
                                ;       R5=BST ADDRESS
                                ;
000134  010446                          MOV     R4,-(SP)
000136                                  CALL    @WN.ROT(R4)             ;CALL ROUTINE
000142  012604                          MOV     (SP)+,R4
                                ;
000144                                  CALL    TRACE                   ;TRACE TRANSITION
                                ;
000150  000470                          BR      TSTDND                  ;SEE ABOUT NEXT NODE
                                ;
                                ;
                                ; SOMETHING IN SSQ
                                ;
000152  010503                  DATSSQ: MOV     R5,R3                   ;ADDRESS OF SSQ -
000154  062703  000004                  ADD     #B.SSQ,R3               ;  HEAD CELL
000160                                  CALL    SEZLOK                  ;SIZE QUEUE
000164  016504  000046                  MOV     B.CNTX(R5),R4           ;STATE TABLE START FOR -
000170  062704  000002                  ADD     #2,R4                   ;  CURRENT NODE
                                ;R5=BST ADDRESS
                                ;R4=STATE TABLE ENTRY ADDRESS
                                ;R3=SSQ HEAD CELL ADDRESS
```

```
                                        ;
000174  005764  000000         NXTENT: TST     WN.SRC(R4)              ;CONTINUE IF NOT -
000200  100003                         BPL     1$                      ;  END OF NODE
000202                                 CALL    RLSLOK                  ;RELEASE QUEUE
000206  000717                        BR      WAITSQ                  ;WAIT FOR SSQ ENTRY
                                        ;
000210  016501  000002         1$:     MOV     B.SSQ-2(R5),R1          ;COUNT OF SSQ ENTRIES
000214  011302                 2$:     MOV     (R3),R2                 ;NEXT SSQ PACKET
000216  026264  000002  000000         CMP     2(R2),WN.SRC(R4)        ;BRANCH IF -
000224  001406                         BEQ     SSQMAT                  ;  COMMAND MATCH
000226  010203                         MOV     R2,R3                   ;SSQ PACKET ADDRESS
000230  077107                         SOB     R1,2$                   ;LOOP THROUGH SSQ ENTRIES
                                    ;ALL SSQ ENTRIES PROCESSED - NO MATCH
000232  011303                         MOV     (R3),R3                 ;R3 POINTS TO HEAD CELL AGAIN
000234  062704  000010                 ADD     #WN.SIZ,R4              ;NEXT NODE ENTRY ADDRESS
000240  000755                         BR      NXTENT                  ;CHECK NEXT NODE ENTRY BY SSQ
                                        ;
                                    ; FOUND A SSQ MATCH
000242  016465  000006  000046 SSQMAT: MOV     WN.NXT(R4),B.CNTX(R5)   ;NEXT NODE
000250                                 CMKT$S                          ;CANCEL TIME OUT
000256  042765  100000  000044         BIC     #BIT15,B.STAT(R5)
                                    ; DEQUEUE PACKET
000264  011213                         MOV     (R2),(R3)               ;LINK AROUND REMOVED PACKET
000266  026502  000006                 CMP     B.SSQ+2(R5),R2          ;BRANCH IF NOT REMOVING
000272  001002                         BNE     1$                      ;  BOTTOM ENTRY OF SSQ
000274  010365  000006                 MOV     R3,B.SSQ+2(R5)          ;NEW BOTTOM ENTRY
000300  005365  000002         1$:     DEC     B.SSQ-2(R5)             ;ONE LESS ENTRY
000304  010503                         MOV     R5,R3                   ;GET HEAD CELL ADDRESS
000306  062703  000004                 ADD     #B.SSQ,R3
000312                                 CALL    RLSLOK                  ;RELEASE QUEUE
                                        ;
                                        ;
                                    ; CALL WAIT NODE CONDITION ROUTINE
                                    ;   R2=SSQ PACKET ADDRESS
                                    ;   R5=BST ADDRESS
                                        ;
000316  010446                         MOV     R4,-(SP)                ;SAVE R4
000320                                 CALL    @WN.ROT(R4)             ;MATCH ROUTINE
000324  012604                         MOV     (SP)+,R4                ;RESTORE R4
                                        ;
000326                                 CALL    TRACE                   ;TRACE THIS NODE
                                        ;
                                    ; CHECK TYPE OF NEXT NODE
                                        ;
000332  026427  000004  177777 TSTDND: CMP     WN.NTP(R4),#DN          ;BRANCH IF DECISION NODE
000340  001413                         BEQ     DESCND
                                        ;
                                    ; GOING TO WAIT NODE - RELEASE PACKET (IF NOT A TIMEOUT)
000342  032765  100000  000044         BIT     #BIT15,B.STAT(R5)       ;NOT A TIME OUT PATH
000350  001404                         BEQ     1$
000352  042765  100000  000044         BIC     #BIT15,B.STAT(R5)       ;CLEAR TIMEOUT
000360  000612                        BR      NEXTST
000362                         1$:     CALL    PUTFRE                  ;RETURN PACKET
000366  000607                        BR      NEXTST                  ;CHECK NEXT NODE AGAINST SSQ
                                        ;
                                        ;
                                    ; A DECISION NODE IS NEXT - CHECK DECISION ROUTINE
```

```
                                        ;
000370  016504  000046                  DESCND: MOV     B.CNTX(R5),R4           ;START OF DECISION NODE
                                        ;
000374  005764  000000                  1$:     TST     DN.DCK(R4)              ;END OF NODE?
000400  100002                                  BPL     2$                      ;NO
000402                                          CALL    CRASH                   ;YES - CAN'T HAPPEN
000406  010446                          2$:     MOV     R4,-(SP)                ;SAVE R4
000410                                          CALL    @DN.DCK(R4)             ;CALL DECISION ROUTINE
000414  012604                                  MOV     (SP)+,R4                ;RESTORE R4
000416  103403                                  BCS     3$                      ;MATCHED
000420  062704  000010                          ADD     #DN.SIZ,R4              ;NO MATCH - TRY NEXT ENTRY
000424  000763                                  BR      1$
                                        ;
000426  016465  000006  000046          3$:     MOV     DN.NXT(R4),B.CNTX(R5)   ;MATCHED - SET NEXT NODE
                                        ;
                                        ;
                                        ; CALL DECISION SATISFIED ROUTINE
                                        ;   R2=SSD PACKET ADDRESS (UNLESS FOLLOWING A TIMEOUT)
                                        ;   R5=BST ADDRESS
                                        ;
000434  010446                                  MOV     R4,-(SP)                ;SAVE R4
000436                                          CALL    @DN.ROT(R4)             ;DECISION SATISFIED ROUTINE
000442  012604                                  MOV     (SP)+,R4                ;RESTORE R4
                                        ;
000444                                          CALL    TRACE                   ;TRACE THIS NODE
                                        ;
000450  000730                                  BR      TSTDND                  ;CHECK NEXT NODE TYPE
                                        ;
                                        ;
```

```
                        .SBTTL  STATE TABLE MACROS
;
;
; STATE TRANSITION TABLE MACROS
;
; WNODE: GENERATE A WAIT NODE ENTRY
;
; PARMS: COMMAND SOURCE, COMMAND TYPE, RPOUTINE ADDRESS, DN OR UN,
; NEXT STATE ADDRESS
;
.MACRO  WNODE   A,B,C,D,E
        .BYTE   A       ;COMMAND SOURCE
        .BYTE   B       ;COMMAND TYPE
        .WORD   C       ;ROUTINE ADDRESS
        .WORD   D       ;NEXT NODE TYPE
        .WORD   E       ;NEXT STATE
.ENDM   WNODE
;
;
; TIMOUT: GENERATE A TIMEOUT ENTRY FOR WAIT NODE
;
; PARMS: TIMEOUT-SATISFIED ROUTINE, DN OR UN, NEXT STATE ADDRESS
;
.MACRO  TIMOUT  A,B,C
        .WORD   -1      ;TIMEOUT INDICATOR
        .WORD   A       ;ROUTINE ADDRESS
        .WORD   B       ;NEXT NODE TYPE
        .WORD   C       ;NEXT STATE
.ENDM   TIMOUT
;
;
; TIME: GENERATE TIMEOUT VALUE FOR A WAIT NODE
;
; PARM: WAIT TIME IN SECONDS (0 MEANS NO TIMEOUT)
;
.MACRO  TIME    A
        .WORD   A
.ENDM   TIME
;
;
;
; DNODE: GENERATE A DECISION NODE ENTRY
;
; PARMS: DECISION CHECK ROUTINE ADDRESS, DECISION-SATISFIED ROUTINE
; ADDRESS, DN OR UN, NEXT STATE ADDRESS
;
.MACRO  DNODE   A,B,C,D
        .WORD   A       ;DECISION CHECK ROUTINE
        .WORD   B       ;ROUTINE ADDRESS
        .WORD   C       ;NEXT NODE TYPE
        .WORD   D       ;NEXT STATE
.ENDM   DNODE
;
; NTERM - TERMINATE A DECISION NODE
;
.MACRO  NTERM
        .WORD   -1
```

```
236                             .ENDM   NTERM
237                             ;
238                             ;
```

```
                        .SBTTL  STATE TRANSITION TABLES
                        ;
                        ; ENTRY FROM SCHEDULER ROOT TASK
                        ;
                        ; WAITING FOR A QUQ ENTRY OR AN EXTERNAL COMMAND
                        ;
000452                  TOPNOD:
000452                  TIME    5*60.
000454                  WNODE   XMSCHED,0,STTINT,WN,WTSSRC      ;INIT SAYS TO INITIALIZE
000464                  WNODE   XBATCH,1,CNOP,DN,BCHEMP         ;CLOSE COMMAND FROM CONSOLE
000474                  WNODE   XMSCHED,4,TULQUQ,DN,QUQENE      ;QUQ ENTRY MADE
000504                  WNODE   XMSCHED,1,SSUXDN,DN,BCHEMP      ;SUX% DONE
000514                  WNODE   XMSCHED,2,SPRDON,WN,TOPNOD      ;SEARCH DONE
000524                  TIMOUT  CNOP,DN,SHRTCL                  ;5 MIN T/O - SEE ABOUT SHORT CYCLE
                        ;
                        ; SEE IF WE CAN DO A SHORT CYCLE
                        ;
000534                  SHRTCL:
000534                  DNODE   CBEMPT,TULQUQ,DN,QUQENE         ;BATCH NOT EMPTY - GO TO QUQ
000544                  DNODE   SGISIN,STRTSC,DN,GOCDBU         ;SEARCH GO IS IN - DO SHORT CYCLE
000554                  DNODE   TRUE,TULQUQ,DN,QUQENE           ;ELSE GO TO QUQ
000564                  NTERM
                        ;
                        ; SEE IF THERE IS ANYTHING IN THE QUQ
                        ;
000566                  QUQENE:
000566                  DNODE   QUQEMP,CNOP,WN,TOPNOD           ;QUQ EMPTY NOW
000576                  DNODE   CMDBQU,CNOP,DN,SMDBUR           ;GOT MDBU REQUEST
000606                  DNODE   TRUE,PRBUHL,WN,UPRBRS           ;GOT QUQ - PROBE UHL SIZE
000616                  NTERM
                        ;
                        ; MDBU REQUEST IN - SEE IF BATCH EMPTY
                        ;
000620                  SMDBUR:
000620                  DNODE   CBESTT,STMDBU,DN,CSRGMD         ;BATCH EMPTY - DO MDBU
000630                  DNODE   TRUE,ROMDBR,WN,QT0TCL           ;NOT EMPTY - REQUEUE MDBU, CLOSE
000640                  NTERM
                        ;
                        ; CONSOLE BATCH CLOSE COMMAND IN -
                        ; SX% DONE IN - CLOSE BATCH IF NOT EMPTY
                        ;
000642                  BCHEMP:
000642                  DNODE   CBESTT,TULQUQ,DN,QUQENE         ;BATCH EMPTY - GO TO QUQ
000652                  DNODE   TRUE,BTCHCL,WN,QT0TCL           ;BATCH NOT EMPTY
000662                  NTERM
                        ;
                        ; QUQ ENTRY IN - SEE IF HLMERG HAS ROOM FOR UHL
                        ;
000664                  UPRBRS:
000664                  TIME    15.
000666                  WNODE   XMSCHED,4,CNOP,WN,UPRBRS        ;PURGE THESE SSQ ENTRIES
000676                  WNODE   XHLMERG,2,MRGQRY,WN,QT0RSP      ;UHL WILL FIT
000706                  WNODE   XHLMERG,3,CNOP,DN,RMBIGQ        ;UHL WILL NOT FIT
000716                  TIMOUT  PRTMSG,WN,UPRBRS
                        ;
                        ; GAVE QUERY TO QT0 TO MERGE
                        ;
```

```
 59 000726                      QT0RSP:
 60 000726                      TIME    15.
 61 000730                      WNODE   XQT0.0,MRGUHL,WN,HLMBOR        ;QUERY MERGED - BATCH OPEN
 62 000740                      WNODE   XQT0.1,CNOP,DN,RMBIGQ           ;QUERY NOT MERGED (SIZE) - BATCH OPEN
 63 000750                      WNODE   XQT0.3,MRGUHL,WN,HLMBCR         ;QUERY MERGED - BATCH CLOSE
 64 000760                      WNODE   XQT0.4,GRIDOR,DN,QUQENE         ;QUERY NOT MERGED (PARSE) - BATCH OPEN
 65 000770                      WNODE   XQT0.5,DLRORM,WN,QTABTR         ;BATCH CORRUPTED
 66 001000                      TIMOUT  PRTMSG,WN,QT0RSP
 67                             ;
 68                             ; GAVE UHL TO HLMERG; QT0 HAD NOT CLOSED
 69                             ;
 70 001010                      HLMBOR:
 71 001010                      TIME    15.
 72 001012                      WNODE   XHLMERG,0,CNOP,DN,OBCDES        ;UHL MERGED - SEE ABOUT OTHER
 73                                                                     ;  BATCH CLOSURE CONDITIONS
 74 001022                      TIMOUT  PRTMSG,WN,HLMBOR
 75                             ;
 76                             ; QUERY AND UHL MERGED; NO BATCH CLOSEURE FROM QT0 OR HLMERG -
 77                             ; SEE ABOUT OTHER BATCH CLOSURE CONDITIONS
 78                             ;
 79 001032                      OBCDES:
 80 001032                      DNODE   CMXQRY,BTCHCL,WN,QT0TCL         ;MAX QUERIES IN BATCH
 81 001042                      DNODE   CSXDIN,BTCHCL,WN,QT0TCL         ;SXX IN WHILE BATCH WAS EMPTY
 82 001052                      DNODE   TRUE,TULQUQ,DN,QUQENE           ;NO CLOSE - TRY NEXT QUQ ENTRY
 83 001062                      NTERM
 84                             ;
 85                             ; QT0 TOLD TO CLOSE DUE TO EXTERNAL CLOSE DECISION
 86                             ;
 87 001064                      QT0TCL:
 88 001064                      TIME    15.
 89 001066                      WNODE   XQT0.2,HLMECL,WN,HLTCLS         ;QT0 CLOSED, NOW CLOSE UHL
 90 001076                      WNODE   XQT0.6,REPRCB,WN,QTABCR         ;BATCH CORRUPTED
 91 001106                      TIMOUT  PRTMSG,WN,QT0TCL
 92                             ;
 93                             ; QT0 ACCEPTED QUERY AND CLOSED BATCH  - GAVE UHL TO HLMERG
 94                             ;
 95 001116                      HLMBCR:
 96 001116                      TIME    15.
 97 001120                      WNODE   XHLMERG,0,HLMECL,WN,HLTCLS      ;TELL HLMERG TO CLOSE
 98 001130                      TIMOUT  PRTMSG,WN,HLMBCR
 99                             ;
100                             ; HLMERG TOLD TO CLOSE
101                             ;
102 001140                      HLTCLS:
103 001140                      TIME    15.
104 001142                      WNODE   XHLMERG,1,GOTQTS,WN,QTSRSP      ;UHL CLOSED - NOW XLATE
105 001152                      TIMOUT  PRTMSG,WN,HLTCLS
106                             ;
107                             ; TRANSLATOR HAS BEEN RUN
108                             ;
109 001162                      QTSRSP:
110 001162                      TIME    90.
111 001164                      WNODE   XQTS,0,GTCSDS,DN,ONXTBT         ;XLATE SUCCESS
112 001174                      WNODE   XQTS,1,REPRCB,WN,QTABCR         ;BATCH OVERFLOW
113 001204                      TIMOUT  PRTMSG,WN,QTSRSP
114                             ;
115                             ; QTS HAS TRANSLATED - SEE IF NEXT BATCH CAN BE OPENED
```

```
                                ;
001214                          ONXTBT:
001214                          DNODE   NXTBIN,RUNXTS,DN,CSGOIN         ;NEXT BATCH INACTIVE
001224                          DNODE   TRUE,SETACT,DN,CSGOIN           ;NEXT BATCH STILL ACTIVE
001234                          NTERM
                                ;
                                ; THIS BATCH CLOSED - SEE IF SEARCH-GO ALREADY IN
                                ;
001236                          CSGOIN:
001236                          DNODE   SGISIN,GSMODE,WN,CTLODE         ;SEARCH GO IN - SEARCH
001246                          DNODE   TRUE,CNOP,WN,WAITSG             ;NOT IN - WAIT
001256                          NTERM
                                ;
                                ; WAIT FOR PREVIOUS BATCH SEARCH COMPLETION
                                ;
001260                          WAITSG:
001260                          TIME    5*60.
001262                          WNODE   XMSCHED,2,GSMODE,WN,CTLODE      ;GOT SEARCH GO
001272                          TIMOUT  PRTMSG,WN,WAITSG
                                ;
                                ; SEARCH UNIT LOAD STARTED FOR ALL SEARCH UNITS - WAIT
                                ; FOR SULOAD TO REPORT IN FOR ALL SU
                                ;
001302                          CTLODE:
001302                          TIME    60.
001304                          WNODE   XSULOAD,0,SSULOD,DN,ALODED      ;ONE SULOAD REPORTING SU LOADED
001314                          TIMOUT  PRTMSG,WN,CTLODE
                                ;
                                ; SEE IF ALL SEARCH UNITS LOADED
                                ;
001324                          ALODED:
001324                          DNODE   CASULD,CNOP,WN,WTSXPD           ;ALL SEARCH UNITS LOADED
001334                          DNODE   TRUE,CNOP,WN,CTLODE             ;SOME NOT LOADED - WAIT
001344                          NTERM
                                ;
                                ; ALL SEARCH UNITS LOADED - WAIT FOR SUX% DONE
                                ;
001346                          WTSXPD:
001346                          TIME    5*60.
001350                          WNODE   XDMCIN,0,SXPDIN,DN,AXPDIN       ;ONE SUX% DONE IN
001360                          TIMOUT  PRTMSG,WN,WTSXPD
                                ;
                                ; SEE IF ALL SUX% DONE ARE IN
                                ;
001370                          AXPDIN:
001370                          DNODE   CAXPDI,CLSNXB,WN,WTSFOS         ;ALL IN - CLOSE NEXT BATCH
001400                          DNODE   TRUE,CNOP,WN,WTSXPD             ;SOME NOT IN YET
001410                          NTERM
                                ;
                                ; ALL SEARCH UNITS SUX% DONE - WAIT FOR FOS
                                ;
001412                          WTSFOS:
001412                          TIME    5*60.
001414                          WNODE   XDMCIN,1,SFOSIN,DN,AFOSIN       ;ONE FOS REPORTED IN
001424                          TIMOUT  PRTMSG,WN,WTSFOS
                                ;
                                ; SEE IF ALL FOS IN
```

```
                                ;
001434                          AFOSIN:
001434                          DNODE   CAFOSI,MRGFOS,WN,FOSMRS         ;ALL IN - MERGE FOS
001444                          DNODE   TRUE,CNOP,WN,WTSFOS             ;SOME FOS NOT YET IN
001454                          NTERM
                                ;
                                ; WAIT FOR FOSMRG TO COMPLETE
                                ;
001456                          FOSMRS:
001456                          TIME    30.
001460                          WNODE   XFOSMRG,1,DELCTB,DN,GOCDBU      ;FOS MERGED - DELETE TABLES - CDBU
001470                          TIMOUT  PRTMSG,WN,FOSMRS
                                ;
                                ; IN CDBU - RUN DBPROC
                                ;
001500                          GOCDBU:
001500                          DNODE   TRUE,DBPCDB,WN,DBCRSP           ;RUN DBPROC
001510                          NTERM
                                ;
                                ; CDBU STARTED - GET 'NO WORK' FROM DBPROC, OR 'DATA-BASE END' FROM DMCIN
                                ;
001512                          DBCRSP:
001512                          TIME    3*60.
001514                          WNODE   XDBPROC,0,GSRECR,WN,WTSSRC      ;NO UPDATES - GET STATUS RECORDS
001524                          WNODE   XDMCIN,2,SCDBUD,DN,ACDBUD       ;ONE SU DONE WITH CDBU
001534                          TIMOUT  PRTMSG,WN,DBCRSP
                                ;
                                ; ONE SU REPORTED CDBU DONE - SEE IF ALL DONE
                                ;
001544                          ACDBUD:
001544                          DNODE   CACDBU,GSRECR,WN,WTSSRC         ;ALL DONE - GET STATUS RECORDS
001554                          DNODE   TRUE,CNOP,WN,DBCRSP             ;ALL NOT DONE - WAIT
001564                          NTERM
                                ;
                                ; STATUS RECORDS REQUESTED - WAIT FOR RECORDS TO COME IN
                                ;
001566                          WTSSRC:
001566                          TIME    30.
001570                          WNODE   XDMCIN,3,SSRCIN,DN,ASRCIN       ;ONE STATUS RECORD IN
001600                          TIMOUT  PRTMSG,WN,WTSSRC
                                ;
                                ; SEE IF ALL STATUS RECORDS ARE IN
                                ;
001610                          ASRCIN:
001610                          DNODE   CASRCI,WRAPUP,0,0               ;ALL IN - WRAP UP
001620                          DNODE   TRUE,CNOP,WN,WTSSRC             ;SOME NOT IN YET
001630                          NTERM
                                ;
                                ;
```

```
                        .SBTTL  MDBU STATES
                        ;
                        ; MDBU CAN BE STARTED - SEE IF SEARCH GO ALREADY IN
                        ;
001632                  CSRGMD:
001632                  DNODE   SGISIN,MDBUGO,WN,DBMRSP         ;SEARCH-GO ALREADY IN - INIT MDBU
001642                  DNODE   TRUE,CNOP,WN,WSRCGO             ;WAIT FOR SEARCH-GO
001652                  NTERM
                        ;
                        ; MDBU CAN BE STARTED - WAIT FOR PREVIOUS BATCHES TO RUN DOWN
                        ;
001654                  WSRCGO:
001654                  TIME    5*60.
001656                  WNODE   XMSCHED,2,MDBUGO,WN,DBMRSP      ;GOT SEARCH GO - INIT MDBU
001666                  TIMOUT  PRTMSG,WN,WSRCGO
                        ;
                        ; MDBU IN PROGRESS - WAIT FOR 'DATA BASE END' FROM DMCIN
                        ;
001676                  DBMRSP:
001676                  TIME    0
001700                  WNODE   XDMCIN,2,SCDBUD,DN,AMDBUD       ;GOT A SU TERMINATION
001710                  TIMOUT  CNOP,WN,DBMRSP
                        ;
                        ; ONE SU REPORTED MDBU DONE - SEE IF ALL DONE
                        ;
001720                  AMDBUD:
001720                  DNODE   CACDBU,MDBUDN,DN,ACDBUD         ;ALL SU DONE WITH MDBU - GET SREC
001730                  DNODE   TRUE,CNOP,WN,DBMRSP             ;SOME SU NOT DONE - WAIT
001740                  NTERM
                        ;
```

```
                                .SBTTL  BATCH FAILURE PATHS
                               ;
                               ; **** QT0 REJECTED QUERY AS TOO BIG
                               ;
                               ; QUERY TOO BIG TO FIT IN BATCH - SEE IF ONLY QUERY
                               ;
001742                         RMBIGQ:
001742                         DNODE   SINGLQ,DLQRAP,DN,QUQENE         ;DUMP SINGLE QUERY - GO TO QUQ
001752                         DNODE   TRUE,RQQTRM,WN,QT0TCL           ;REQUEUE LAST QUERY - TERM BATCH
001762                         NTERM
                               ;
                               ; **** QT0 COULDN'T MERGE QUERY - BATCH CORRUPTED
                               ;
                               ; DUMPED LAST QUERY; REQUED REST - QT0 TOLD TO ABORT
                               ;
001764                         QTABTR:
001764                         TIME    15.
001766                         WNODE   XQT0,7,ABRTHL,WN,ABHLRS         ;QT0 ABORTED - ABORT HLMERG
001776                         TIMOUT  PRTMSG,WN,QTABTR
                               ;
                               ; HLMERG TOLD TO ABORT
                               ;
002006                         ABHLRS:
002006                         TIME    15.
002010                         WNODE   XHLMERG,4,TULQUQ,DN,QUQENE      ;HLMERG ABORTED - RESTART BATCH
002020                         TIMOUT  PRTMSG,WN,ABHLRS
                               ;
                               ; **** CORRUPTED BATCH DETECTED BY QT0 ON TERMINATE OR
                               ; BY QTS ON BATCH TRANSLATE
                               ;
                               ; QT0 TOLD TO ABORT
                               ;
002030                         QTABCR:
002030                         TIME    15.
002032                         WNODE   XQT0,7,ABRTHL,WN,ABHLCR         ;QT0 ABORTED - NOW HLMERG
002042                         TIMOUT  PRTMSG,WN,QTABCR
                               ;
                               ; HLMERG TOLD TO ABORT
                               ;
002052                         ABHLCR:
002052                         TIME    15.
002054                         WNODE   XHLMERG,4,CNOP,DN,DWNTSQ        ;HLMERG ABORTED - SINGLE Q LEFT?
002064                         TIMOUT  PRTMSG,WN,ABHLCR
                               ;
                               ; SEE IF DOWN TO LAST QUERY IN BATCH
                               ;
002074                         DWNTSQ:
002074                         DNODE   SINGLQ,DLQRAP,DN,QUQENE         ;DUMP QUERY - CONTINUE EMPTY BATCH
002104                         DNODE   TRUE,RQLRPR,WN,RPRQT0           ;REQUEUE LAST - REPROCESS BATCH
002114                         NTERM
                               ;
                               ; QUERY GIVEN TO QT0 TO MERGE
                               ;
002116                         RPRQT0:
002116                         TIME    15.
002120                         WNODE   XQT0,0,GIVUHL,WN,RPRHLM         ;QT0 MERGED - MERGE UHL
002130                         TIMOUT  PRTMSG,WN,RPRQT0
```

```
311                                     ;
312                                     ; UHL GIVEN TO HLMERG TO MERGE
313                                     ;
314 002140                              RPRHLM:
315 002140                              TIME    15.
316 002142                              WNODE   XHLMERG,0,CNOP,DN,SALORP        ;UHL MERGED - SEE IF ENTIRE
317                                                                             ;             BATCH REPROCESSED
318 002152                              TIMOUT  PRTMSG,WN,RPRHLM
319                                     ;
320                                     ; SEE IF ENTIRE BATCH REPROCESSED
321                                     ;
322 002162                              SALORP:
323 002162                              DNODE   TAORPR,BTCHCL,WN,QT0TCL         ;ALL REPROCESSED - TERM BATCH
324 002172                              DNODE   TRUE,CONTRP,WN,RPRQT0           ;ENTIRE BATCH NOT REPROCESSED - CONT
325 002202                              NTERM
326                                     ;
```

```
                                .SBTTL  DECISION ROUTINES
                        ;
                        ; CALLED IN A DECISION NODE
                        ; OUTPUT:  CARRY SET - DECISION SATISFIED
                        ;          CARRY CLEAR - DECISION NOT SATISFIED
                        ;
002204  000261          TRUE:   SEC
002206                          RTN
002210  000241          FALSE:  CLC
002212                          RTN
                        ;
                        ;
                        ; TEST IF QUQ EMPTY
005765  000056          QUQEMP: TST     B.QUQP(R5)              ;TEST IF QUQ ENTRY WAS UNLOADED
001771                          BEQ     TRUE                    ;EMPTY
000772                          BR      FALSE                   ;GOT QUQ ENTRY
                        ;
                        ;
                        ; TEST IF QUQ ENTRY IS MDBU REQUEST
005765  000060          CMDBQU: TST     B.CQUQ(R5)              ;TEST UNLOADED QUQ ENTRY
100765                          BMI     TRUE
000766                          BR      FALSE
                        ;
                        ;
                        ; CHECK IF MAX ENTRIES IN BATCH
026527  000232  000031  CMXQRY: CMP     B.NQRY(R5),#N.QURY
103360                          BHIS    TRUE
000761                          BR      FALSE
                        ;
                        ;
                        ; CHECK IF SUXX DONE CAME IN WHILE BATCH WAS EMPTY
032765  000001  000044  CSXDIN: BIT     #BIT0,B.STAT(R5)
001353                          BNE     TRUE
000754                          BR      FALSE
                        ;
                        ;
                        ; SEE IF BATCH EMPTY
005765  000232          CBESTT: TST     B.NQRY(R5)
001747                          BEQ     TRUE                    ;BATCH EMPTY
000750                          BR      FALSE
                        ;
                        ;
                        ; SEE IF BATCH NOT EMPTY
005765  000232          CBEMPT: TST     B.NQRY(R5)
001343                          BNE     TRUE                    ;BATCH NOT EMPTY
000744                          BR      FALSE
                        ;
                        ;
                        ; NOW IN CLOSED STATE - SEE IF NEXT BATCH IS ACTIVE
116500  000052          NXTBIN: MOVB    B.NMBR(R5),R0           ;CURRENT BATCH NUMBER
062700  000002                  ADD     #2,R0                   ;NEXT BATCH NUMBER
020027  000006                  CMP     R0,#N.BHGH              ;ZERO IF BATCH
101401                          BLOS    1$                      ;  NUMBER WRAP-AROUND
005000                          CLR     R0
016001  000000G         1$:    MOV     BSTPTR(R0),R1           ;NEXT BATCH
122761  000000  000053          CMPB    #BS.INA,B.STTE(R1)      ;BRANCH IF
001326                          BNE     FALSE                   ;  NOT INACTIVE
```

```
 59 002334  000723                          BR      TRUE
 60                                 ;
 61                                 ;
 62                                 ; TEST IF WE HAVE ALREADY GOT A SEARCH-GO FROM THE PREVIOUS BATCH
 63 002336  032765  000002  000044  SGISIN: BIT     #BIT1,B,STAT(R5)
 64 002344  001317                          BNE     TRUE
 65 002346  000720                          BR      FALSE
 66                                 ;
 67                                 ;
 68                                 ; SEE IF ALL SU LOADED
 69 002350  012701  000001          CASULD: MOV     #N.SUNT-1,R1            ;HIGH SU #
 70 002354  006301                  1$:     ASL     R1                      ;BRANCH IF SOME SU
 71 002356  016100  000000G                 MOV     SUINDX(R1),R0           ;  STILL NOT LOADED
 72 002362  006201                          ASR     R1
 73 002364  022760  000002  000000          CMP     #SU.SRC,SS.STT(R0)
 74 002372  001306                          BNE     FALSE
 75 002374  005301                          DEC     R1                      ;TRY ALL SU
 76 002376  100366                          BPL     1$
 77 002400  000701                          BR      TRUE                    ;ALL LOADED
 78                                 ;
 79                                 ;
 80                                 ; SEE IF ALL SUX% DONE ARE IN
 81 002402  012701  000001          CAXPDI: MOV     #N.SUNT-1,R1            ;HIGH SU#
 82 002406  006301                  1$:     ASL     R1                      ;BRANCH IF SOME SU
 83 002410  016100  000000G                 MOV     SUINDX(R1),R0           ;  STILL NOT SENT SUX% DONE
 84 002414  006201                          ASR     R1
 85 002416  022760  000003  000000          CMP     #SU.XPD,SS.STT(R0)
 86 002424  001271                          BNE     FALSE
 87 002426  005301                          DEC     R1                      ;TRY ALL SU
 88 002430  100366                          BPL     1$
 89 002432  000664                          BR      TRUE                    ;ALL ARE IN
 90                                 ;
 91                                 ;
 92                                 ; SEE IF ALL FOS ARE IN
 93 002434  012701  000001          CAFOSI: MOV     #N.SUNT-1,R1            ;HIGH SU#
 94 002440  006301                  1$:     ASL     R1
 95 002442  016100  000000G                 MOV     SUINDX(R1),R0           ;SU ENTRY ADDRESS
 96 002446  006201                          ASR     R1
 97 002450  022760  000004  000000          CMP     #SU.DBU,SS.STT(R0)      ;BRANCH IF SOME
 98 002456  001254                          BNE     FALSE                   ;  SU NOT DONE
 99 002460  005301                          DEC     R1                      ;TRY ALL SU
100 002462  100366                          BPL     1$
101 002464  000647                          BR      TRUE
102                                ;
103                                ;
104                                ; CHECK IF ALL SU ARE DONE WITH CDBU OR MDBU
105 002466  012701  000001          CACDBU: MOV     #N.SUNT-1,R1            ;HIGH SU #
106 002472  006301                  1$:     ASL     R1
107 002474  016100  000000G                 MOV     SUINDX(R1),R0           ;SU ENTRY ADDRESS
108 002500  006201                          ASR     R1
109 002502  022760  000005  000000          CMP     #SU.SRR,SS.STT(R0)      ;BRANCH IF SOME SU NOT
110 002510  001237                          BNE     FALSE                   ;  YET DONE
111 002512  005301                          DEC     R1                      ;TRY ALL SU
112 002514  100366                          BPL     1$
113 002516  000632                          BR      TRUE                    ;ALL SU DONE
114                                ;
115                                ;
```

```
                                        ; SEE.IF.ALL.STATUS.RECORDS.ARE.IN.
002520  012701  030001          CASRCI: MOV.     #N.SUNT-1.R1            ;HIGH.SU#.
002524  006301                  1$:     ASL.     R1                      ;BRANCH.IF.SOME.SU.
002526  016100  000000G.                MOV.     SUINDX(R1).R0           ;  STILL.NOT.SENT.SREC.
002532. 006201                          ASR.     R1
002534  022760  000006  000000          CMP.     #SU.DON.SS.STT(R0)
002542. 001222.                         BNE.     FALSE.
002544  005301                          DEC.     R1                      ;TRY.ALL.SU.
002546  100366                          BPL.     1$
002550  000615                          BR       TRUE                    ;ALL.ARE.IN.
                                        ;
                                        ;
                                        ; SEE.IF.EXACTLY.ONE.QUERY.IN.BATCH.
002552. 022765  000001  000232. SINGLQ: CMP.     #1.B.NQRY(R5)
002560  001611                          BEQ.     TRUE                    ;ONE.QUERY
002562. 000612.                         BR       FALSE.
                                        ;
                                        ;
                                        ; SEE.IF.ALL.QUERIES.HAVE.BEEN.REPROCESSED.FOR.CORRUPTED.BATCH.
002564  016546  000232          TAQRPR: MOV.     B.NQRY(R5).-(SP)        ;LAST.QUERY'S.ID.
002570  005316                          DEC.     (SP)                    ;  IN.BATCH.
002572. 022662. 000006                  CMP.     (SP)+.6(R2)             ;COMPARE.AGAINST.CURRENT.QUERY.ID.
002576  001602.                        BEQ.     TRUE
002600  000603                          BR       FALSE.
                                        ;
                                        ;
```

```
                                        .SBTTL  CONDITION MATCH ROUTINES
                                ;
                                ; AT ALL TIMES: R5=BST ADDRESS
                                ;               R2=UNLOADED PACKET
                                ;
                                ;
                                ;
                                ; INITIALIZE SCHEDULER ON FIRST BATCH
002602  112765  000004  000053  STTINT: MOVB    #BS.DBU,B.STTE(R5)      ;BATCH STATUS
002610  005065  000054                 CLR     B.BSTA(R5)
                                ;
002614                                  CALL    RUNXTS                  ;RUN NEXT SCHEDULER
002620                                  CALL    CLSNXB                  ;CLOSE NEXT BATCH
                                ;
002624                                  CALL    HSTSRT                  ;START QUERIES FROM HOST
                                ;
002630                                  CALL    GSRECR                  ;GET THE STATUS RECORDS
                                ;
002634                                  RTN
                                ;
                                ;
                                ; TRY TO UNLOAD QUQ
002636                          TULQUQ: CALL    GETQUQ                  ;GET TOP QUQ ENTRY
002642                                  RTN
                                ;
                                ;
                                ; SUX% DONE HAS COME IN
002644  052765  000001  000044  SSUXDN: BIS     #BIT0,B.STAT(R5)        ;SET INDICATOR
002652                                  RTN
                                ;
                                ;
                                ; SEARCH GO CAME IN WHILE BATCH WAS EMPTY
002654  052765  000002  000044  SPRDON: BIS     #BIT1,B.STAT(R5)        ;SET INDICATOR
002662                                  RTN
                                ;
                                ;
                                ; BUILD QUERY SPOOL FILE ENTRY - CALL HLMERG TO SEE IF UHL WILL FIT
002664  016500  000232          PRBUHL: MOV     B.NQRY(R5),R0           ;GET CURRENT # OF QUERIES
002670  005265  000232                  INC     B.NQRY(R5)              ;ONE MORE QUERY IN BST
002674  010001                          MOV     R0,R1                   ;SAVE IT
002676  060000                          ADD     R0,R0                   ;INDEX INTO EQID MAP
002700  060500                          ADD     R5,R0
002702  016560  000064  000234          MOV     B.CQUQ+4(R5),B.QMAP(R0) ;MOVE IN EQID
                                ;
002710  070127  000014                  MUL     #Q.SIZE,R1              ;INDEX INTO SPOOL AREA
002714  060501                          ADD     R5,R1
002716  062701  000316                  ADD     #B.QSPL,R1
002722  016521  000060                  MOV     B.CQUQ(R5),(R1)+        ;# QUERY BLOCKS
002726  016521  000062                  MOV     B.CQUQ+2(R5),(R1)+      ;# UHL BLOCKS
002732  016521  000066                  MOV     B.CQUQ+6(R5),(R1)+      ;FDSC
002736  016521  000070                  MOV     B.CQUQ+8.(R5),(R1)+
002742  016521  000072                  MOV     B.CQUQ+10.(R5),(R1)+
002746  005065  000056                  CLR     B.QUOP(R5)              ;MARK CURRENT QUQ ENTRY UNLOADED
                                ;
002752  112765  000002  000013          MOVB    #2,B.SNDP+1(R5)         ;SEND PROBE COMMAND
002760  116565  000052  000014          MOVB    B.NMBR(R5),B.SNDP+2(R5) ;  TO HLMERG
002766  016565  000232  000016          MOV     B.NQRY(R5),B.SNDP+4(R5)
```

```
 59 002774  005365  000016                 DEC     B.SNDP+4(R5)            ;ZERO ORGIN QID
 60 003000  012700  000002                 MOV     #XHLMERG,R0
 61 003004                                 CALL    RUNTSK
 62 003010                                 RTN
 63                                 ;
 64                                 ;
 65                                 ; HLMERG SAYS UHL WILL FIT - MERGE QUERY
 66 003012  112765  000000  000013 MRQRY: MOVB    #0,B.SNDP+1(R5)         ;SEND MERGE QUERY
 67 003020  116565  000052  000014        MOVB    B.NMBR(R5),B.SNDP+2(R5) ;  COMMAND TO QT0
 68 003026  016565  000232  000016        MOV     B.NQRY(R5),B.SNDP+4(R5)
 69 003034  005365  000016                 DEC     B.SNDP+4(R5)            ;ZERO ORGIN QID
 70 003040  012700  000001                 MOV     #XQT0,R0
 71 003044                                 CALL    RUNTSK
 72 003050                                 RSUM$S  R0                      ;QT0 SUSPENDS
 73 003064                                 RTN
 74                                 ;
 75                                 ;
 76                                 ; QT0 MERGED QUERY; BATCH NOT FULL - MERGE UHL
 77 003066  112765  000000  000013 MRGUHL: MOVB    #0,B.SNDP+1(R5)         ;MERGE UHL COMMAND
 78 003074  116565  000052  000014        MOVB    B.NMBR(R5),B.SNDP+2(R5) ;  TO HLMERG
 79 003102  016565  000232  000016        MOV     B.NQRY(R5),B.SNDP+4(R5)
 80 003110  005365  000016                 DEC     B.SNDP+4(R5)            ;ZERO ORGIN QID
 81 003114  012700  000002                 MOV     #XHLMERG,R0
 82 003120                                 CALL    RUNTSK
 83 003124                                 RTN
 84                                 ;
 85                                 ;
 86                                 ; MDBU REQUEST CAME IN WHILE BATCH WAS NON-EMPTY -
 87                                 ; REQUEUE REQUEST TO QUQ, CLOSE BATCH
 88 003126                         ROMDBR: SAVE    R2                      ;EXISTING PACKET ADDRESS
 89 003130                                 CALL    GETFRE                  ;GET NEW PACKET
 90 003134  012702  177777  000002        MOV     #-1,2(R2)               ;MDBU REQUEST
 91 003142                                 CALL    PUTQUT                  ;REQUEUE TO TOP
 92 003146                                 RESTOR  R2
 93                                 ;
 94 003150  005065  000056                 CLR     B.QUQP(R5)              ;MARK QUQ ENTRY PROCESSED
 95                                 ;
 96 003154  000400                         BR      BTCHCL                  ;CLOSE THE BATCH
 97                                 ;
 98                                 ;
 99                                 ; BATCH CLOSE DECISION MADE - TELL QT0 AND HLMERG TO TERMINATE
100 003156  112765  000001  000013 BTCHCL: MOVB    #1,B.SNDP+1(R5)         ;CLOSE COMMAND TO QT0
101 003164  116565  000052  000014        MOVB    B.NMBR(R5),B.SNDP+2(R5)
102 003172  012700  000001                 MOV     #XQT0,R0
103 003176                                 CALL    RUNTSK
104 003202                                 RSUM$S  R0                      ;QT0 SUSPENDS
105 003216                                 RTN
106                                ;
107                                ;
108                                ; NOW TELL HLMERG TO CLOSE
109 003220  112765  000001  000013 HLMECL: MOVB    #1,B.SNDP+1(R5)         ;CLOSE COMMAND TO HLMERG
110 003226  116565  000052  000014        MOVB    B.NMBR(R5),B.SNDP+2(R5)
111 003234  012700  000002                 MOV     #XHLMERG,R0
112 003240                                 CALL    RUNTSK
113 003244                                 RTN
114                                ;
115                                ;
```

```
                                        ; QT0 AND HLMERG HAVE CLOSED - GO TO OPEN-TRANSLATE STATE AND
                                        ; RUN THE TRANSLATOR
003246  012765  000001  000054  GOTQTS: MOV     #1,B.BSTA(R5)           ;OPEN-XLATE STATE
                                        ;
003254  112765  000000  000013          MOVB    #0,B.SNDP+1(R5)         ;RUN THE TRANSLATOR
003262  116565  000052  000014          MOVB    B.NMBR(R5),B.SNDP+2(R5)
003270  012700  000003                  MOV     #XQTS,R0
003274                                  CALL    RUNTSK
003300                                  RTN
                                        ;
                                        ;
                                        ; TRANSLATOR HAS RUN - GO TO CLOSED STATE
003302  112765  000002  000053  GTCSDS: MOVB    #BS.CLS,B.STTE(R5)
003310  005065  000054                  CLR     B.BSTA(R5)
003314                                  RTN
                                        ;
                                        ;
                                        ; IN CLOSED STATE - RUN NEXT SCHED TO OPEN BATCH
003316  116500  000052          RUNXTS: MOVB    B.NMBR(R5),R0           ;GENERATE NEXT BATCH NUMBER
003322  062700  000002                  ADD     #2,R0
003326  020027  000006                  CMP     R0,#N.BHGH
003332  101401                          BLOS    1$
003334  005000                          CLR     R0
003336                          1$:     CALL    RUNSCH                  ;RUN SCHEDULER
003342                                  RTN
                                        ;
                                        ;
                                        ; ALL BATCHES ACTIVE - SET FLAG, TELL HOST TO STOP
003344  052765  000004  000044  SETACT: BIS     #BIT2,B.STAT(R5)
003352                                  CALL    HSTSTP                  ;TELL HOST TO STOP
003356                                  RTN
                                        ;
                                        ;
                                        ; GO TO SEARCH MODE - LOAD CONTROL TABLES
003360  032765  000004  000044  GSMODE: BIT     #BIT2,B.STAT(R5)        ;BRANCH IF ALL BATCHES NOT ACTIVE
003366  001407                          BEQ     1$
003370  042765  000004  000044          BIC     #BIT2,B.STAT(R5)        ;OK NOW
003376                                  CALL    RUNXTS                  ;OPEN A BATCH
003402                                  CALL    HSTSRT                  ;TELL HOST TO START
                                        ;
003406  112765  000003  000053  1$:     MOVB    #BS.SRC,B.STTE(R5)      ;SEARCH IS CURRENT STATE
003414  005065  000054                  CLR     B.BSTA(R5)
003420  052765  000010  000044          BIS     #BIT3,B.STAT(R5)        ;SHOW WE ARE DOING A SEARCH
                                        ;
                                        ; GENERATE BATCH CUTOFF MSG
003426  016500  000010          BCOFMS: MOV     B.SFDB(R5),R0           ;FDB ADDRESS
003432  016004  000022                  MOV     F.BKDS+2(R0),R4         ;BUFFER ADDRESS
003436  012724  041103                  MOV     #"CB,(R4)+              ;EXCHANGE ID
003442  005724                          TST     (R4)+                   ;PAD WORD
003444  116524  000052                  MOVB    B.NMBR(R5),(R4)+        ;BATCH NUMBER
003450  105024                          CLRB    (R4)+
003452  016503  000232                  MOV     B.NQRY(R5),R3           ;NUMBER OF QUERIES
003456  010324                          MOV     R3,(R4)+
                                        ;
003460  010500                          MOV     R5,R0                   ;ADDRESS OF FIRST EQID
003462  062700  000234                  ADD     #B.QMAP,R0
003466  005001                          CLR     R1                      ;QID 0
```

```
003470  012024                  1$:     MOV     (R0)+,(R4)+             ;MOVE IN EQID
003472  010124                          MOV     R1,(R4)+                ;MOVE IN QID
003474  005201                          INC     R1                      ;NEXT QID
003476  077304                          SOB     R3,1$                   ;LOOP
                                ;BUILD SPOOL FILE - SEND TO HOTSK
003500  112765  000003  000013          MOVB    #3,B.SNDP+1(R5)         ;COMMAND
003506  010504                          MOV     R5,R4                   ;FDSC IN PACKET
003510  062704  000014                  ADD     #B.SNDP+2,R4
003514                                  CALL    WRTSPL                  ;WRITE FILE
003520  012700  000010                  MOV     #XHOTSK,R0              ;SEND PACKET TO HOTSK
003524                                  CALL    RUNTSK
003530                                  RSUM$S  R0                      ;HOTSK SUSPENDS
                                ;
                                ; DELETE QUERY SPOOL FILES
003544  016500  000010          DELOSP: MOV     B.SFDB(R5),R0           ;FDB ADDRESS
003550  016503  000232                  MOV     B.NQRY(R5),R3           ;# OF SPOOL FILES
003554  001412                          BEQ     2$                      ;NONE
                                ;
003556  010501                          MOV     R5,R1                   ;FIRST FDSC ADDRESS
003560  062701  000000C                 ADD     #B.QSPL+0,FDSC,R1
                                ;
003564                          1$:     CALL    BLDEFL                  ;BUILD FNB
003570                                  CALL    .DLFNB                  ;DELETE FILE
003574  062701  000014                  ADD     #Q.SIZE,R1              ;NEXT FDSC
003600  077307                          SOB     R3,1$                   ;LOOP FOR ALL FILES
                                ;
                                ;
003602  012703  000001          2$:     MOV     #N.SUNT-1,R3            ;HIGH SEARCH UNIT #
003606  112765  000000  000013  3$:     MOVB    #0,B.SNDP+1(R5)         ;COMMAND TO SULOAD
003614  116565  000052  000014          MOVB    B.NMBR(R5),B.SNDP+2(R5)
003622  010365  000016                  MOV     R3,B.SNDP+4(R5)         ;SEARCH UNIT TO LOAD
003626  012700  000005                  MOV     #XSULOAD,R0
003632                                  CALL    RUNTSK                  ;RUN SULOAD
003636  006303                          ASL     R3                      ;SEARCH UNIT STATUS-
003640  016304  000000G                 MOV     SUINDX(R3),R4           ;   TABLE ENTRY
003644  006203                          ASR     R3
003646  012764  000001  000000          MOV     #SU.LOD,SS.STT(R4)      ;SET STATUS TO LOAD
003654  005303                          DEC     R3                      ;FOR ALL SEARCH UNITS
003656  100353                          BPL     3$
                                ;
003660  116567  000052  177776G          MOVB    B.NMBR(R5),SUST-2       ;BATCH NUMBER BEING SEARCHED
003666                                  RTN
                                ;
                                ;
                                ; SULOAD REPORTED A SEARCH UNIT LOADED - UPDATE ITS STATUS
003670  016200  000006          SSULOD: MOV     6(R2),R0                ;SEARCH UNIT NUMBER
003674  060000                          ADD     R0,R0
003676  016000  000000G                 MOV     SUINDX(R0),R0           ;SUST ENTRY ADDRESS
003702  012760  000002  000000          MOV     #SU.SRC,SS.STT(R0)      ;UPDATE SU STATUS
003710                                  RTN
                                ;
                                ;
                                ; SUX% DONE REPORTED IN - UPDATE SU STATUS
003712  016200  000004          SXPDIN: MOV     4(R2),R0                ;SU #
003716  060000                          ADD     R0,R0
003720  016000  000000G                 MOV     SUINDX(R0),R0           ;SUST ENTRY ADDRESS
003724  012760  000003  000000          MOV     #SU.XPD,SS.STT(R0)      ;SET SU STATUS
```

```
230 003732                              RTN
231                                 ;
232                                 ;
233                                 ; ALL SUX% DONE ARE IN - CLOSE NEXT BATCH
234 003734                          CLSNXB: SAVE    R2                      ;CURRENT PACKET
235 003736                                  CALL    GETFRE                  ;GET A PACKET
236 003742  112762  000000  000002          MOVB    #XMSCHED,2(R2)          ;GO TO SCHEDULER
237 003750  112762  000001  000003          MOVB    #1,3(R2)                ;COMMAND
238 003756  116500  000052                  MOVB    B.NMBR(R5),R0           ;GET NEXT SCHEDULER BATCH NUMBER
239 003762  062700  000002                  ADD     #2,R0
240 003766  020027  000006                  CMP     R0,#N.BHGH
241 003772  101401                          BLOS    1$
242 003774  005000                          CLR     R0
243 003776  010062  000004          1$:     MOV     R0,4(R2)                ;BATCH NUMBER
244 004002                                  CALL    PUTSSQ                  ;QUEUE PACKET
245 004006                                  RESTOR  R2                      ;CURRENT PACKET
246                                 ;
247 004010                                  RTN
248                                 ;
249                                 ;
250                                 ; FOS REPORTED IN - UPDATE SU STATUS
251 004012  016200  000004          SFOSIN: MOV     4(R2),R0                ;SEARCH UNIT #
252 004016  060000                          ADD     R0,R0
253 004020  016000  000000G                 MOV     SUINDX(R0),R0           ;SUST ENTRY ADDRESS
254 004024  012760  000004  000000          MOV     #SU.DBU,SS.STT(R0)      ;SET SU STATUS
255 004032                                  RTN
256                                 ;
257                                 ;
258                                 ; TELL FOSMRG TO MERGE
259 004034  112765  000000  000013  MRGFOS: MOVB    #0,B.SNDP+1(R5)         ;MERGE COMMAND
260 004042  116565  000052  000014          MOVB    B.NMBR(R5),B.SNDP+2(R5) ;BATCH NUMBER
261 004050  012700  000007                  MOV     #XFOSMRG,R0
262 004054                                  CALL    RUNTSK
263 004060                                  RTN
264                                 ;
265                                 ;
266                                 ; FOS MERGED - DELETE CONTROL FILES, GO TO DBU STATE
267 004062  016500  000010          DELCTB: MOV     B.SFDB(R5),R0           ;FDB ADDRESS
268 004066  012703  004126'                 MOV     #BFDSPT,R3              ;TABLE OF FILES TO DELETE
269                                 ;
270 004072  012301                  1$:     MOV     (R3)+,R1                ;FDSC INDEX
271 004074  100006                          BPL     2$                      ;CONTINUE IF NOT END OF TABLE
272                                 ;
273 004076  112765  000004  000053          MOVB    #BS.DBU,B.STTE(R5)      ;BATCH IN DBU STATE
274 004104  005065  000054                  CLR     B.BSTA(R5)              ;CDBU
275 004110                                  RTN
276                                 ;
277 004112  060501                  2$:     ADD     R5,R1                   ;FDSC ADDRESS
278 004114                                  CALL    BLDEFL                  ;BUILD FNB
279 004120                                  CALL    .DLFNB                  ;DELETE FILE
280 004124  000762                          BR      1$                      ;GO AGAIN
281                                 ;
282                                 ; TABLE OF FILES TO DELETE
283 004126                          BFDSPT:
284 004126  000132                          .WORD   B.FEMA
285 004130  000142                          .WORD   B.FEMB
286 004132  000152                          .WORD   B.FEMC
```

```
                                        .WORD   B.FQLS
                                        .WORD   B.FNHR
                                        .WORD   B.FFSA
                                        .WORD   B.FFSB
                                        .WORD   B.FFSC
                                        .WORD   -1
                                ;
                                ;
                                ; START SHORT CYCLE
004150  012701  000001          STRTSC: MOV     #N,SUNT-1,R1            ;PUT ALL SU IN DBU STATE
004154  006301                  1$:     ASL     R1
004156  016100  000000G                 MOV     SUINDX(R1),R0
004162  006201                          ASR     R1
004164  012760  000004  000000          MOV     #SU.DBU,SS.STT(R0)
004172  005301                          DEC     R1
004174  100367                          BPL     1$
                                ;
004176  112765  000004  000053          MOVB    #BS.DBU,B.STTE(R5)      ;BATCH IN CDBU STATE
004204  005065  000054                  CLR     B.BSTA(R5)
                                ;
004210                                  CALL    RUNXTS                  ;OPEN NEXT BATCH
004214                                  CALL    CLSNXB                  ;REQUEST NEXT BATCH TO CLOSE
                                ;
004220                                  RTN
                                ;
                                ;
                                ; RUN DBPROC TO DO CDBU
004222  112765  000000  000013  DBPCDB: MOVB    #0,B.SNDP+1(R5)         ;COMMAND
004230  116565  000052  000014          MOVB    B.NMBR(R5),B.SNDP+2(R5) ;BATCH NUMBER
004236  012700  000012                  MOV     #XDBPROC,R0             ;TO DBPROC
004242                                  CALL    RUNTSK
004246                                  RTN
                                ;
                                ;
                                ; ONE SU REPORTED CDBU DONE - UPDATE SU STATUS
004250  016200  000004          SCDBUD: MOV     4(R2),R0                ;SU #
004254  060000                          ADD     R0,R0
004256  016000  000000G                 MOV     SUINDX(R0),R0           ;SU TABLE ENTRY ADDRESS
004262  012760  000005  000000          MOV     #SU.SRR,SS.STT(R0)      ;NEW STATUS
004270                                  RTN
                                ;
                                ;
                                ; RUN GTSREC TO GET THE STATUS RECORDS
004272  012703  000001          GSRECR: MOV     #N,SUNT-1,R3            ;HIGH SU#
004276  112765  000000  000013  1$:     MOVB    #0,B.SNDP+1(R5)         ;COMMAND TO GTSREC
004304  010365  000014                  MOV     R3,B.SNDP+2(R5)         ;SU # TO REQUEST
004310  012700  000014                  MOV     #XGTSREC,R0             ;RUN GTSREC
004314                                  CALL    RUNTSK
004320  006303                          ASL     R3                      ;SUST ENTRY ADDRESS
004322  016304  000000G                 MOV     SUINDX(R3),R4
004326  006203                          ASR     R3
004330  012764  000005  000000          MOV     #SU.SRR,SS.STT(R4)      ;SET SU STATUS
004336  005303                          DEC     R3                      ;FOR ALL SU
004340  100356                          BPL     1$
                                ;
004342                                  RTN
                                ;
```

```
                                        ;
                                        ;STATUS RECORD REPORTED IN - UPDATE SU STATUS
004344  016200  000004          SSRCIN: MOV     4(R2),R0                ;SU #
004350  060000                          ADD     R0,R0
004352  016000  000000G                  MOV     SUINDX(R0),R0           ;SUST ENTRY ADDRESS
004356  012760  000006  000000          MOV     #SU.DON,SS.STT(R0)      ;NEW STATUS
004364                                  RTN
                                        ;
                                        ;
                                        ; SEARCH CYCLE COMPLETE - WRAP-UP THIS SCHEDULER
                                        ;
004366                                  WRAPUP:
004366  032765  000010  000044          BIT     #BIT3,B.STAT(R5)        ;BRANCH IF DIDN'T DO
004374  001402                          BEQ     STATDN                  ;  A SEARCH
004376                                  CALL    DOSTAT                  ;ACCUMULATE CURRENT STATS
                                        ; MAKE OUR SEARCH UNITS INACTIVE
004402  012701  000001          STATDN: MOV     #N.SUNT-1,R1            ;LOOP THROUGH SU'S
004406  006301                  1$:     ASL     R1
004410  016100  000000G                  MOV     SUINDX(R1),R0
004414  006201                          ASR     R1
004416  012760  000000  000000          MOV     #SU.IDL,SS.STT(R0)      ;IDLE
004424  005301                          DEC     R1
004426  100367                          BPL     1$
004430  112767  177777  177776G        MOVB    #-1,SUST-2              ;NO BATCH BEING SEARCHED
                                        ; TELL NEXT BATCH TO SEARCH
004436                          NXBCSH: ALTP$S  ,#100.                  ;HIGH PRI DURING WRAPUP
                                        ;
004454                                  SAVE    R2                      ;OLD PACKET
004456                                  CALL    GETFRE                  ;GET A PACKET
004462  112762  000000  000002          MOVB    #XMSCHED,2(R2)          ;GO TO NEXT SCHEDULER
004470  112762  000002  000003          MOVB    #2,3(R2)
004476  116500  000052                  MOVB    B.NMBR(R5),R0           ;GET NEXT SCHED'S BATCH NUMBER
004502  062700  000002                  ADD     #2,R0
004506  020027  000006                  CMP     R0,#N.BHGH
004512  101401                          BLOS    1$
004514  005000                          CLR     R0
004516  010062  000004          1$:     MOV     R0,4(R2)                ;BATCH NUMBER
004522                                  CALL    PUTSSQ                  ;QUEUE PACKET
004526                                  RESTOR  R2                      ;OLD PACKET
                                        ; GET RID OF CURRENT SSQ PACKET
004530                          DMPSSQ: CALL    PUTFRE                  ;RELEASE PACKET
                                        ; PURGE SSQ
004534  010503                          MOV     R5,R3                   ;GET TOP SSQ PACKET
004536                                  CALL    GETSSQ
004542  005702                          TST     R2                      ;BRANCH IF GOT ONE
004544  001371                          BNE     DMPSSQ
                                        ; CLEAN UP OUR BST
004546  112765  000000  000053          MOVB    #BS.INA,B.STTE(R5)      ;BATCH INACTIVE
004554  005065  000054                  CLR     B.BSTA(R5)
004560  005065  000044                  CLR     B.STAT(R5)              ;SCHEDULER STATUS FLAGS
004564  005065  000232                  CLR     B.NQRY(R5)              ;NO QUERY SPOOL FILES
                                        ;
                                        ; EXIT SCHEDULER
004570                                  EXIT$S
                                        ;
                                        ;
                                        ;
```

```
                                ;
                                ; NO OPERATION
004576                          CNOP:   RTN
                                ;
                                ;
                                ; CRASH CAN'T-HAPPEN CASE
004600  000167  177775          CRASH:  JMP     .+1
                                ;
                                ;
```

```
                                        .SBTTL  MDBU CONDITION MATCH ROUTINES
                                ;
                                ;
                                ; START A MDBU REQUEST
004604                          STMDBU: CALL    HSTSTP                  ;TELL HOST TO STOP QUERIES
                                ;
004610  012701  000001                  MOV     #N,SUNT-1,R1            ;PUT SU'S IN DBU STATE
004614  006301                  1$:     ASL     R1
004616  016100  000000G                 MOV     SUINDX(R1),R0
004622  006201                          ASR     R1
004624  012760  000004  000000          MOV     #SU.DBU,SS.STT(R0)
004632  005301                          DEC     R1
004634  100367                          BPL     1$
                                ;
004636  112765  000004  000053          MOVB    #BS.DBU,B.STTE(R5)      ;BATCH IN MDBU STATE
004644  012765  000001  000054          MOV     #BIT0,B.BSTA(R5)
                                ;
004652                                  RTN
                                ;
                                ;
                                ; ACHIEVED BATCH RUN-DOWN - START DBPROC DOING MDBU
004654  112765  000001  000013  MDBUGO: MOVB    #1,B.SNDP+1(R5)         ;RUN DBPROC FOR MDBU
004662  116565  000052  000014          MOVB    B.NMBR(R5),B.SNDP+2(R5)
004670  012700  000012                  MOV     #XDBPROC,R0
004674                                  CALL    RUNTSK
                                ;
004700  112765  000002  000013          MOVB    #2,B.SNDP+1(R5)         ;TELL HOST TO START MDBU
004706  012700  000010                  MOV     #XHOTSK,R0
004712                                  CALL    RUNTSK
004716                                  RSUM$S  R0                      ;HOTSK SUSPENDS
004732                                  RTN
                                ;
                                ;
                                ; MDBU COMPLETED - START QUERIES, OPEN AND CLOSE NEXT BATCH
004734                          MDBUDN: CALL    HSTSRT                  ;START QUERIES
004740                                  CALL    RUNXTS                  ;OPEN NEXT BATCH
004744                                  CALL    CLSNXB                  ;REQUEST NEXT BATCH CLOSE
004750                                  RTN
                                ;
```

```
                                        .SBTTL  ERROR PATH CONDITION ROUTINES
                                ;
                                ;
                                ; DUMP SINGLE QUERY IN BATCH - CONTINUE WITH EMPTY BATCH
004752  005001                  DLQRAP: CLR     R1                      ;QID ZERO
004754  012703  000144                  MOV     #QE.RO1,R3              ;ERROR CODE
004760                          MSGHDP: CALL    PARSEM                  ;MESSAGE TO HOST
                                ;
004764                                  CALL    DMPQRY                  ;DUMP QUERY
004770  000167  175642                  JMP     TULQUQ                  ;GO TO QUQ
                                ;
                                ;
                                ; DUMP LAST QUERY IN BATCH (PARSE ERROR) - CONTINUE WITH OPEN BATCH
004774  016201  000006          GRIDQR: MOV     6(R2),R1                ;QID
005000  016203  000010                  MOV     8.(R2),R3               ;ERROR CODE
005004  000765                          BR      MSGHDP                  ;MESSAGE TO HOST, DUMP QUERY
                                ;
                                ;
                                ; REQUEUE LAST QUERY - TERMINATE BATCH
005006                          RQQTRM: CALL    REQQRY
005012  000167  176140                  JMP     BTCHCL
                                ;
                                ;
                                ; DUMP LAST QUERY - REQUEUE REMAINING QUERIES IN BATCH
005016  016201  000006          DLRQRM: MOV     6(R2),R1                ;QID
005022  016203  000010                  MOV     8.(R2),R3               ;ERROR CODE
005026                                  CALL    PARSEM                  ;MESSAGE TO HOST
005032                                  CALL    DMPQRY                  ;DUMP LAST QUERY
                                ;
005036  005765  000232          1$:     TST     B.NQRY(R5)              ;REQUEUE REMAINING
005042  001403                          BEQ     ABRTQT
005044                                  CALL    REQQRY
005050  000772                          BR      1$
                                ;
                                ;
                                ; ABORT TO QT0 FOR CURRENT BATCH
005052  112765  000002  000013  ABRTQT: MOVB    #2,B.SNDP+1(R5)         ;ABORT COMMAND
005060  116565  000052  000014          MOVB    B.NMBR(R5),B.SNDP+2(R5) ;BATCH #
005066  012700  000001                  MOV     #XQT0,R0
005072                                  CALL    RUNTSK
005076                                  RSUM$S  R0                      ;QT0 SUSPENDS
005112                                  RTN
                                ;
                                ;
                                ; ABORT T0 HLMERG FOR CURRENT BATCH
005114  112765  000003  000013  ABRTHL: MOVB    #3,B.SNDP+1(R5)         ;ABORT COMMAND
005122  116565  000052  000014          MOVB    B.NMBR(R5),B.SNDP+2(R5) ;BATCH #
005130  012700  000002                  MOV     #XHLMERG,R0
005134                                  CALL    RUNTSK
005140                                  RTN
                                ;
                                ;
                                ; START TO REPROCESS CORRUPTED BATCH
005142  005065  000054          REPRCB: CLR     B.BSTA(R5)              ;BATCH STATE OPEN, ACTIVE
                                ; DELETE CONTROL FILES
005146  016500  000010                  MOV     B.SFDB(R5),R0           ;R0-> FDB
005152  012703  004126'                 MOV     #BFDSPT,R3              ;R3->TABLE OF FILES TO DELETE
```

```
                                1$:     MOV     (R3)+,R1                ;R1 = FDSC INDEX
                                        BMI     2$                      ;BRANCH IF DONE
                                        ADD     R5,R1                   ;R1->FDSC
                                        CALL    BLDEFL                  ;BUILD FNB
                                        CALL    .DLFNB                  ;DELETE FILE
                                        BR      1$                      ;GO AGAIN
                                2$:     BR      ABRTQT
                                ;
                                ;
                                ; SEVERAL QUERIES IN BATCH - REQUEUE LAST ONE AND REPROCESS REST
                                ROLRPR: CALL    REQQRY                  ;REQUE LAST QUERY
                                ;
                                        CLR     B,SNDP+4(R5)            ;START WITH QUERY #0
                                NXTQRP: MOVB    #0,B,SNDP+1(R5)         ;MERGE COMMAND
                                        MOVB    B,NMBR(R5),B,SNDP+2(R5) ;BATCH NUMBER
                                        MOV     #XQT0,R0
                                        CALL    RUNTSK
                                        RSUM$S  R0
                                        RTN
                                ;
                                ;
                                ; QT0 MERGED QUERY - GIVE UHL TO HLMERG
                                GIVUHL: MOV     6(R2),B,SNDP+4(R5)      ;QID
                                        MOVB    #0,B,SNDP+1(R5)         ;COMMAND TO MERGE
                                        MOVB    B,NMBR(R5),B,SNDP+2(R5) ;BATCH #
                                        MOV     #XHLMERG,R0
                                        CALL    RUNTSK
                                        RTN
                                ;
                                ;
                                ; ALL QUERIES NOT REPROCESSED YET - GO WITH NEXT ONE
                                CONTRP: MOV     6(R2),B,SNDP+4(R5)      ;NEXT QID
                                        INC     B,SNDP+4(R5)
                                        BR      NXTQRP                  ;GO TO QT0
                                ;
```

```
                                        .SBTTL  STATE TABLE SUPPORT ROUTINES
                                ;
                                ;
                                ; ISSUE SEND DIRECTIVE FOR SCEDULER AND RUN TASK
                                ;
                                ; INPUT: DATA IN B.SNDP
                                ;        R0=DESTINATION MODULE ID
                                ; OUTPUT: DATA SENT, MODULE REQUESTED
                                ;
005322  060000                  RUNTSK: ADD     R0,R0                   ;GENERATE ADDRESS OF MODULE NAME
005324  060000                          ADD     R0,R0
005326  062700  005420'                 ADD     #MNINDX,R0
005332  062705  000012                  ADD     #B.SNDP,R5              ;ADDRESS OF DATA
005336                                  SDAT$S  R0,R5                   ;SEND THE DATA
                                ;
005356                                  RQST$S  R0                      ;RUN THE TASK
005402  005025                          CLR     (R5)+                   ;CLEAR BEGINNING OF SEND PACKET
005404  005025                          CLR     (R5)+
005406  005025                          CLR     (R5)+
005410  005025                          CLR     (R5)+
005412  162705  000022                  SUB     #B.SNDP+8.,R5           ;RESTORE BST
005416                                  RTN
                                ;
                                ;
                                ; MODULE NAMES
                                ;
005420  052073  031314          MNINDX: .RAD50  /MSCHED/                ;0
005424  066576  000000                  .RAD50  /DT0   /                ;1
005430  031755  021027                  .RAD50  /HLMERG/                ;2
005434  066563  000000                  .RAD50  /QTS   /                ;3
005440  014534  056754                  .RAD50  /DBLOAD/                ;4
005444  075024  056754                  .RAD50  /SULOAD/                ;5
005450  015413  035160                  .RAD50  /DMCIN /                ;6
005454  023753  052027                  .RAD50  /FOSMRG/                ;7
005460  032154  074170                  .RAD50  /HOTSK /                ;8
005464  031574  074170                  .RAD50  /HITSK /                ;9
005470  014540  071333                  .RAD50  /DBPROC/                ;10
005474  006274  012000                  .RAD50  /BATCH /                ;11
005500  027363  070513                  .RAD50  /GTSREC/                ;12
                                ;
                                ;
                                ;
                                ; OPEN A NEW BATCH
                                ;
                                ; INPUT: R0 = BATCH NUMBER FOR SCHEDULER
                                ;
005504  060000                  RUNSCH: ADD     R0,R0                   ;GENERATE ADDRESS OF SCHD NAME
005506  062700  005540'                 ADD     #SCHNAM,R0
005512                                  RQST$S  R0                      ;RUN PROPER SCHEDULER
005536                                  RTN
                                ;
                                ;
005540  073500  016716          SCHNAM: .RAD50  /SCHD00/
005544  073500  016720                  .RAD50  /SCHD02/
005550  073500  016722                  .RAD50  /SCHD04/
005554  073500  016724                  .RAD50  /SCHD06/
                                ;
```

```
                                ;
                                ; TELL HOST TO START/STOP QUERIES
                                ;
                                ; INPUT: NONE
                                ;
005560  112765  000001  000013 HSTSTP: MOVB    #1.B.SNDP+1(R5)         ;STOP QUERIES
005566  000403                 BR      HSTSND
                                ;
005570  112765  000000  000013 HSTSRT: MOVB    #0.B.SNDP+1(R5)         ;START QUERIES
005576  012700  000010         HSTSND: MOV     #XHOTSK,R0              ;GO TO HOTSK
005602                                 CALL    RUNTSK
005606                                 RSUM$S  R0                      ;HOTSK SUSPENDS
005622                                 RTN
                                ;
                                ;
```

```
                                        ;
                                        ; ACCUMULATE BATCH STATISTICS FOR CURRENT HOUR
                                        ;
005624                          DOSTAT: SAVE    R2
005626  012704  000000G                MOV     #CHSTAT,R4              ;ADDRESS OF STATS
                                        ;
005632  005264  000000                  INC     ST.BTC(R4)              ;ONE MORE BATCH
005636  066564  000232  000002          ADD     B.NQRY(R5),ST.QRY(R4)   ;NUMBER OF QUERIES
005644  016501  000110                  MOV     B.UDOC(R5),R1           ;NUMBER OF DOC PER UHL
005650  016502  000112                  MOV     B.UDOC+2(R5),R2
005654  016500  000232                  MOV     B.NQRY(R5),R0
005660                                  CALL    $DDIV
005664  060264  000006                  ADD     R2,ST.UHL+2(R4)
005670  005564  000004                  ADC     ST.UHL(R4)
005674  060164  000004                  ADD     R1,ST.UHL(R4)
005700  066564  000116  000012          ADD     B.HDOC+2(R5),ST.HRL+2(R4) ;NUMBER OF HRL DOC
005706  005564  000010                  ADC     ST.HRL(R4)
005712  066564  000114  000010          ADD     B.HDOC(R5),ST.HRL(R4)
005720  066564  000076  000016          ADD     B.QTTM(R5),ST.XLT+2(R4) ;TRANSLATION TIME
005726  005564  000014                  ADC     ST.XLT(R4)
005732  066564  000100  000022          ADD     B.FSAZ(R5),ST.ASZ+2(R4) ;FSA SIZE
005740  005564  000020                  ADC     ST.ASZ(R4)
005744  066564  000102  000026          ADD     B.FSBZ(R5),ST.BSZ+2(R4) ;FSB SIZE
005752  005564  000024                  ADC     ST.BSZ(R4)
005756  066564  000104  000032          ADD     B.FSCZ(R5),ST.CSZ+2(R4) ;FSC SIZE
005764  005564  000030                  ADC     ST.CSZ(R4)
005770  066564  000106  000036          ADD     B.QLSZ(R5),ST.QSZ+2(R4) ;QLS SIZE
005776  005564  000034                  ADC     ST.QSZ(R4)
                                        ;
006002  005001                          CLR     R1                      ;SEARCH TIME PER SU
006004  012703  000002                  MOV     #N.SUNT,R3
006010  010302                  1$:     MOV     R3,R2
006012  005302                          DEC     R2
006014  060202                          ADD     R2,R2
006016  016202  000000G                 MOV     SRECPT(R2),R2
006022  066201  000002                  ADD     SR,SRC(R2),R1
006026  077310                          SOB     R3,1$
006030  005000                          CLR     R0
006032  071027  000002                  DIV     #N.SUNT,R0
006036  060064  000042                  ADD     R0,ST.SCH+2(R4)
006042  005564  000040                  ADC     ST.SCH(R4)
                                        ;
006046                                  RESTOR  R2
006050                                  RTN
                                        ;
```

```
                                        ;
                                        ; TRACE STATE TABLE CHANGES
                                        ;
                                        ; INPUT - R4= CURRENT NODE ADDRESS
                                        ;         SYSFLG-BIT0= TRACE FLAG
                                        ;
006052  032767  000001  000000G TRACE:  BIT     #BIT0,SYSFLG            ;BRANCH IF TRACING
006060  001001                          BNE     TRCISO
006062                                  RTN                             ;RETURN
                                        ;
006064                                  TRCISO: SAVE    R4
006066  016503  000010                  MOV     B.SFDB(R5),R3           ;BUILD LINE IN FDB - FNB
006072  062703  000102                  ADD     #F.FNB,R3
                                        ;
006076  010401                          MOV     R4,R1                   ;CURRENT NODE ADDRESS
006100  162701  000000'                 SUB     #STTENT,R1
006104  072127  000004                  ASH     #4,R1
006110  012704  000004                  MOV     #4,R4                   ;LOOP COUNT
                                        ;
006114  005000                          1$:     CLR     R0              ;STORE FOUR CHARACTERS
006116  073027  000003                  ASHC    #3,R0
006122  062700  000060                  ADD     #60,R0
006126  110023                          MOVB    R0,(R3)+
006130  077407                          SOB     R4,1$
006132  112723  000040                  MOVB    #' ,(R3)+
                                        ;
006136  116501  003052                  MOVB    B.NMBR(R5),R1           ;BATCH NUMBER
006142                                  CALL    STRTWO
006146  112723  000040                  MOVB    #' ,(R3)+
                                        ;
006152  016504  000010                  MOV     B.SFDB(R5),R4           ;GET TIME
006156  062704  000120                  ADD     #F.FNB+14.,R4
006162                                  GTIM$S  R4
                                        ;
006172  062704  000010                  ADD     #8.,R4                  ;STORE ASCII TIME IN BUFFER
006176  012401                          MOV     (R4)+,R1
006200                                  CALL    STRTWO
006204  112723  000072                  MOVB    #':,(R3)+
006210  012401                          MOV     (R4)+,R1
006212                                  CALL    STRTWO
006216  112723  000072                  MOVB    #':,(R3)+
006222  012401                          MOV     (R4)+,R1
006224                                  CALL    STRTWO
                                        ;
006230                                  RESTOR  R4
                                        ; PRINT LINE
006232  162703  000020                  SUB     #16.,R3
006236                                  QIOW$S  #IO.WLB,#TRCLUN,#1,,,<R3,#16.,#40>
                                        ;
006306                                  RTN
                                        ;
                                        ;
                                        ; STORE TWO DECIMAL DIGITS
                                        ;
                                        ; R1= 2-DIGIT NUMBER
                                        ; R3= PLACE TO STORE IT
                                        ;
```

```
178 006310  005000                .STRTWO:: CLR     R0
179 006312  071027  000012                DIV     #10.,R0
180 006316  062700  000060                ADD     #60.R0          ;HIGH ORDER
181 006322  110023                        MOVB    R0.(R3)+
182 006324  062701  000060                ADD     #60.R1
183 006330  110123                        MOVB    R1.(R3)+        ;LOW ORDER
184 006332                                RTN
185                                 ;
186                                 ;
```

```
                                        ;
                                        ;
                                        ; GENERATE ERROR·EXCHANGE·TO·HOST·
                                        ;
                                        ; INPUT: R1=·QID·
                                        ;        R3=·ERROR·CODE·
                                        ;
006334  016500  000010        PARSEM: MOV·    B.SFDB(R5),R0          ;FDB·ADDRESS·
006340  016004  000022                MOV·    F.BKDS+2(R0),R4        ;BUFFER·ADDRESS·
006344  012724  050505                MOV·    #"EQ,(R4)+             ;EXCHANGE·ID·
006350  005724                        TST·    (R4)+                  ;PAD·
006352· 060101                        ADD·    R1,R1                  ;GET·QID·INDEX·
006354  060501                        ADD·    R5,R1
006356  016124  000234                MOV·    B.QMAP(R1),(R4)+       ;EQID·OF·BAD·QUERY·
006362· 010314                        MOV·    R3,(R4)                ;ERROR·CODE·
                              ;BUILD·SPOOL·FILE·-·SEND·TO·HOTSK·
006364  112765  000004  000013        MOVB·   #4,B.SNDP+1(R5)        ;COMMAND·TO·HOTSK·
006372· 010504                        MOV·    R5,R4                  ;FDSC·IN·PACKET·
006374  062704  000014                ADD·    #B.SNDP+2,R4
006400                                CALL·   WRTSPL·                ;WRITE·FILE·
006404  012700  000010                MOV·    #XHOTSK,R0             ;SEND·PACKET·TO·HOTSK·
006410                                CALL·   RUNTSK·
006414                                RSUM#S· R0                     ;HOTSK·SUSPENDS·
006430                                RTN·
                                        ;
```

```
                                        ;
                                        ; REQUEUE LAST QUERY IN BATCH TO QUO
                                        ;
                                        ; INPUT - R5= BST
                                        ;         B.NQRY= # OF QUERIES IN BATCH
                                        ;
006432                                  REQQRY: SAVE    R2                      ;SAVE CURRENT SSQ PACKET ADDRESS
006434                                          CALL    GETFRE                  ;GET FREE PACKET
                                        ;
006440  005365  000232                          DEC     B.NQRY(R5)              ;ONE LESS QUERY
006444  016501  000232                          MOV     B.NQRY(R5),R1           ;GENERATE EQID INDEX
006450  010100                                  MOV     R1,R0
006452  060000                                  ADD     R0,R0
006454  060500                                  ADD     R5,R0
006456  016062  000234  000006                  MOV     B.QMAP(R0),6(R2)        ;EQID TO PACKET
                                        ;
006464  070127  000014                          MUL     #Q.SIZE,R1              ;GENERATE SPOOL AREA INDEX
006470  062701  000316                          ADD     #B.QSPL,R1
006474  060501                                  ADD     R5,R1
006476  016162  000000  000002                  MOV     Q.NQBK(R1),2(R2)        ;MOVE SPOOL FILE INFO TO PACKET
006504  016162  000002  000004                  MOV     Q.NUHL(R1),4(R2)
006512  016162  000004  000010                  MOV     Q.FDSC(R1),8.(R2)
006520  016162  000006  000012                  MOV     Q.FDSC+2(R1),10.(R2)
006526  016162  000010  000014                  MOV     Q.FDSC+4(R1),12.(R2)
006534  016162  000012  000016                  MOV     Q.FDSC+6(R1),14.(R2)
                                        ;
006542                                          CALL    PUTQUT                  ;QUO PACKET ON TOP
006546                                          RESTOR  R2                      ;RESTORE CURRENT SSQ ENTRY
006550                                          RTN
                                        ;
                                        ;
                                        ; REMOVE LAST QUERY FROM BATCH
                                        ;
                                        ; INPUT - R5= BST
                                        ;         B.NQRY= # OF QUERIES IN BATCH
                                        ;
006552  005365  000232                  DMPQRY: DEC     B.NQRY(R5)              ;ONE LESS QUERY
006556  016501  000232                          MOV     B.NQRY(R5),R1           ;GENERATE ADDRESS OF FDSC
006562  070127  000014                          MUL     #Q.SIZE,R1
006566  062701  000000C                         ADD     #B.QSPL+Q.FDSC,R1
006572  060501                                  ADD     R5,R1
006574  016500  000010                          MOV     B.SFDB(R5),R0           ;FDB
006600                                          CALL    BLDEFL                  ;BUILD FNB
006604                                          CALL    .DLFNB                  ;DELETE SPOOL FILE
                                        ;
                                                                                ;SPECIAL EXCHANGE TO HOST
                                        ;
006610  012700  006674'                          MOV     #MSG1,R0                ;ERROR MSG TO CONSOLE
006614                                          CALL    COMSG
                                        ;
006620                                          RTN
                                        ;
```

```
                                        ;
                                        ;
                                        ; CONSOLE MESSAGE TO CONSOLE
                                        ;
                                        ; INPUT - R0= MESSAGE ADDRESS
                                        ;
006622  012001                          COMSG:  MOV     (R0)+,R1
006624                                          QIOW$S  #IO.WLB,5,#COLUN,,#1,,,<R0,R1,#40>
006672                                          RTN
                                        ;
                                        ;
                                        ;
                                        ; ERROR MESSAGES
                                        ;
006674  000024                          MSG1:   .WORD   MSG1L
006676      115     123     103         MSG1T:  .ASCII  /MSCHED: QUERY DUMPED/
006701      110     105     104
006704      072     040     121
006707      125     105     122
006712      131     040     104
006715      125     115     120
006720      105     104
        000024                          MSG1L=.-MSG1T
                                                .EVEN
                                        ;
006722  000022                          MSG2:   .WORD   MSG2L
006724      115     123     103         MSG2T:  .ASCII  /MSCHED: DISK ERROR/
006727      110     105     104
006732      072     040     104
006735      111     123     113
006740      040     105     122
006743      122     117     122
        000022                          MSG2L=.-MSG2T
                                                .EVEN
                                        ;
006746  000020                          MSG3:   .WORD   MSG3L
006750      115     123     103         MSG3T:  .ASCII  /MSCHED: TIME-OUT/
006753      110     105     104
006756      072     040     124
006761      111     115     105
006764      055     117     125
006767      124
        000020                          MSG3L=.-MSG3T
                                                .EVEN
                                        ;
```

```
                                        ;
                                        ; WRITE SINGLE-BLOCK-EXCHANGE SPOOL FILE (1024-WORD BLOCK)
                                        ;
                                        ; INPUT - R4= ADDRESS OF WHERE TO BUILD FDSC
                                        ;         DATA IN DATBUF
                                        ;
006770                                  WRTSPL: FDAT$R  B,SFDB(R5),,,#-N,BFAC  ;ALLOCATE FILE
007002  012701  000042                          MOV     #FN,SHD,R1             ;BUILD FNB
007006                                          CALL    BLDNFL
007012                                          OFNB$W                         ;OPEN FILE
007024  103426                                  BCS     FILERR
                                        ;
007026                                          WRITE$                         ;WRITE THE BLOCK
007032                                          WAIT$
007036  105760  000052                          TSTB    F,ERR(R0)
007042  100417                                  BMI     FILERR
                                        ;
007044  016064  000102  000000                  MOV     F,FNB+N,FID(R0),FD,FID(R4) ;MOVE FDSC
007052  016064  000104  000002                  MOV     F,FNB+N,FID+2(R0),FD,FID+2(R4)
007060  016064  000120  000004                  MOV     F,FNB+N,FVER(R0),FD,FVR(R4)
007066  010164  000006                          MOV     R1,FD,FNB(R4)
007072                                          CLOSE$                         ;CLOSE FILE
007076  103401                                  BCS     FILERR
007100                                          RTN
                                        ;
                                        ; ERROR WRITING FILE
                                        FILERR: SAVE    R0
007104                                          SAVE    R1
007106  012700  006722'                        MOV     #MSG2,R0               ;DISK-ERROR MESSAGE
007112                                          CALL    COMSG                  ;MESSAGE TO CONSOLE
007116                                          RESTOR  R1
007120                                          RESTOR  R0
                                        ;
007122                                          CALL    .DLFNB                 ;TRY TO DELETE FILE
007126  005064  000000                          CLR     FD,FID(R4)             ;NO FILE BUILT
007132                                          RTN
                                        ;
```

```
                                        ;
                                        ; PRINT MSG UPON TIMEOUT
                                        ;
007134  012700  006746'         PRTMSG: MOV     #MSG3,R0                ;MESSAGE TO CONSOLE
007140                                  CALL    COMSG
                                        ;
007144                                  CALL    TRCISO                  ;FORCE A TRACE LINE
007150                                  RTN
                                        ;
```

```
                                ;
                                ; TIMEOUT AST
                                ;
007152                          TIMAST: SAVE    R0
007154  116500  000050                  MOVB    B.SSQF(R5),R0           ;SET EVENT FLAG
007160                                  SETF$S  R0
007170                                  RESTOR  R0
                                ;
007172  052765  100000  000044          BIS     #BIT15,B.STAT(R5)       ;TIMEOUT FLAG
                                ;
007200  005726                          TST     (SP)+                   ;REMOVE AST PARAMETER
007202                                  ASTX$S                          ;EXIT AST
                                ;
```

```
1          000001                    .END
```

SYMBOL TABLE

```
ABHLCR  002052R        B.FFSB  000212        010 DBPCDB  004222R        FD.RAN=  000002        F.EOBB=  000032
ABHLRS  002006R        B.FFSC  000222        010 DBSLEN=  000116        FD.REC=  000001        F.ERR =  000052
ABRTHL  005114R        B.FMHR  000172        010 DELCTB  004062R        FD.RWM=  000001        F.FACC=  000043
ABRTQT  005052R        B.FQLS  000162        010 DELQSP  003544R        FD.SDI=  000020        F.FFBY=  000014
ACDBUD  001544R        B.FSAZ  000100        010 DESCND  000370R        FD.SQD=  000040        F.FNAM=  000110
AFOSIN  001434R        B.FSBZ  000102        010 DH.BF0  000002    005 FD.TTY=  000004        F.FNB =  000102
ALODED  001324R        B.FSCZ  000104        010 DH.BF1  000004    005 FD.WBH=  000002        F.FTYP=  000116
AMDBUD  001720R        B.HBLK  000120        010 DH.CTL  000000    005 FF.CHR=  000005        F.FVER=  000120
ASRCIN  001610R        B.HDOC  000114        010 DH.DMC  000010    005 FF.NV =  000003        F.HIBK=  000004
AXPDIN  001370R        B.HRLP  000126        010 DH.FLG  000006    005 FF.POE=  000002        F.LUN =  000042
BCHEMP  000642R        B.HRLR  000122        010 DLQRAP  004752R        FF.RWD=  000001        F.MBCT=  000054
BCOFMS  003426R        B.HRLW  000124        010 DLRQRM  005016R        FF.RWF=  000006        F.MBCI=  000055
BFDSPT  004126R        B.NMBR  000052        010 DMPQRY  006552R        FF.SPC=  000004        F.MBFG=  000056
BITVAL=  000000        B.NQRY  000232        010 DMPSSQ  004530R        FILERR  007102R        F.NRBD=  000024
BIT0  =  000001        B.QLSZ  000106        010 DN    =  177777        FN.DBR  000026    011 F.NREC=  000030
BIT1  =  000002        B.QMAP  000234        010 DN.DCK  000000    013 FN.DBS  000022    011 F.OVBS=  000030
BIT10 =  002000        B.QSPL  000316        010 DN.NTP  000004    013 FN.DHR  000040    011 F.RACC=  000016
BIT11 =  004000        B.QTTM  000076        010 DN.NXT  000006    013 FN.EMA  000012    011 F.RATT=  000001
BIT12 =  010000        B.QUOP  000056        010 DN.ROT  000002    013 FN.EMB  000014    011 F.RCNM=  000034
BIT13 =  020000        B.SFDB  000010        010 DN.SIZ  000010    013 FN.EMC  000016    011 F.RCTL=  000017
BIT14 =  040000        B.SIZE  000772        010 DOSTAT  005624R        FN.FSA  000000    011 F.RSIZ=  000002
BIT15 =  100000        B.SNDP  000012        010 DWNTSQ  002074R        FN.FSB  000002    011 F.RTYP=  000000
BIT2  =  000004        B.SSQ   000004        010 FALSE   002210R        FN.FSC  000004    011 F.SEQN=  000100
BIT3  =  000010        B.SSQF  000050        010 FA.APD=  000100        FN.LGQ  000034    011 F.SPDV=  000072
BIT4  =  000020        B.STAT  000044        010 FA.CRE=  000010        FN.LGU  000036    011 F.SPUN=  000074
BIT5  =  000040        B.STTE  000053        010 FA.DLK=  001000        FN.MFO  000024    011 F.STBK=  000036
BIT6  =  000100        B.UDOC  000110        010 FA.ENB=  100000        FN.MHR  000010    011 F.UNIT=  000136
BIT7  =  000200        CACDBU  002466R            FA.EXC=  002000        FN.NMB  000044    011 F.URBD=  000020
BIT8  =  000400        CAFOSI  002434R            FA.EXT=  000004        FN.QLS  000006    011 F.VBN =  000064
BIT9  =  001000        CASRCI  002520R            FA.NSP=  000100        FN.QRY  000020    011 F.VBSZ=  000060
BLDEFL=  ******  G     CASULD  002350R            FA.POS=  010000        FN.SF0  000030    011 GETFRE=  ******  G
BLDNFL=  ******  G     CAXPDI  002402R            FA.RD =  000001        FN.SF1  000032    011 GETQUO=  ******  G
BSTPTR=  ******  G     CBEMPT  002270R            FA.RWD=  004000        FN.SHD  000042    011 GETSSQ=  ******  G
BS.CLS=  000002        CBESTT  002260R            FA.SEQ=  040000        FOSMRS  001456R        GIVUHL  005252R
BS.DBU=  000004        CF.B0 =  000070            FA.SHR=  020000        FO.APD=  000106        GOCDBU  001500R
BS.INA=  000000        CF.B2 =  000067            FA.TMP=  000020        FO.MFY=  000002        GOTOTS  003246R
BS.OPN=  000001        CF.B4 =  000066            FA.WCK=  020000        FO.RD =  000001        GRIDQR  004774R
BS.SRC=  000003        CF.B6 =  000065            FA.WRT=  000002        FO.UPD=  000006        GSMODE  003360R
BTCHCL  003156R        CF.DR0=  000064            FD.BLK=  000010        FO.WRT=  000016        GSRECR  004272R
BYTE0 =  000000        CF.DR1=  000063            FD.CCL=  000002        F.ACTL=  000076        GTCSDS  003302R
BYTE1 =  000001        CHSTAT=  ******  G         FD.COM=  020000        F.ALOC=  000040        G.TICP=  000016
BYTE2 =  000002        CH.AND=  000001            FD.CR =  000002        F.BBFS=  000062        G.TICT=  000014
BYTE3 =  000003        CLSNXB  003734R            FD.DIR=  000010        F.BDB =  000070        G.TIDA=  000004
BYTE4 =  000004        CMDBQU  002224R            FD.FID  000000    003 F.BGBC=  000057        G.TIHR=  000006
BYTE5 =  000005        CMXQRY  002234R            FD.FNB  000006    003 F.BKDN=  000026        G.TIMI=  000010
BYTE6 =  000006        CNOP    004576R            FD.FTN=  000001        F.BKDS=  000020        G.TIMO=  000002
BYTE7 =  000007        COLUN =  000006            FD.FVR  000004    003 F.BKEF=  000050        G.TISC=  000012
BYTE8 =  000010        COMSG   006622R            FD.FI1=  040000        F.BKP1=  000051        G.TIYR=  000000
BYTE9 =  000011        CONTRP  005306R            FD.INS=  000010        F.BKST=  000024        HLMBCR  001116R
BYTVAL=  000012        CRASH   004600R            FD.ISP=  002000        F.BKVB=  000064        HLMBQR  001010R
B.BSTA  000054    010 CSGOIN  001236R            FD.LEN  000010    003 F.CHR =  000075        HLMECL  003228R
B.CNTX  000046    010 CSRGMD  001632R            FD.MNT=  100000        F.CNTG=  000034        HLTCLS  001140R
B.COUQ  000060    010 CSXDIN  002246R            FD.OSP=  004000        F.DFNB=  000046        HSTSND  005576R
B.FEMA  000132    010 CTLODE  001302R            FD.PLC=  000004        F.DSPT=  000044        HSTSRT  005570R
B.FEMB  000142    010 DATSSQ  000152R            FD.PRN=  000004        F.DVNM=  [redacted]    IO.WLB=  ******  GX
B.FEMC  000152    010 DBCRSP  001512R            FD.PSE=  010000        F.EFBK=  000010        M     =  000052
B.FFSA  000202    010 DBMRSP  001676R            FD.RAH=  000001        F.EFN =  000050
```

```
MDBUDN  004734R        N.PKTS= 000043        SGISIN  002336R        STTINT  002602R          WN.NXT  000006      012
MDBUGO  004654R        N.QURY= 000031        SHRTCL  000534R        ST.ASZ  000020      006 WN.ROT  000002      012
MNINDX  005420R        N.STAT= 000020        SINGLQ  002552R        ST.BSZ  000024      006 WN.SIZ  000010      012
MRGFOS  004034R        N.SUNT= 000002        SMDBUR  000620R        ST.BTC  000000      006 WN.SRC  000000      012
MRGORY  003012R        N.UNIT= 000034        SPRDON  002654R        ST.CSZ  000030      006 WN.TYP  000001      012
MRGUHL  003066R        OBCDES  001032R       SRECPT= ****** G       ST.HRL  000010      006 WORD0 = 000000
MSGHDP  004760R        ONXTBT  001214R       SR.ARE  000114    002 ST.LEN  000044      006 WORD1 = 000002
MSG1    006674R        PARSEM  006334R       SR.ARS  000106    002 ST.QRY  000002      006 WORD2 = 000004
MSG1L = 000024         PAR$$$= 000065        SR.DAY  000010    002 ST.QSZ  000034      006 WORD3 = 000006
MSG1T   006676R        PRBUHL  002664R       SR.DLT  000014    002 ST.SCH  000040      006 WORD4 = 000010
MSG2    005722R        PRTMSG  007134R       SR.ECB  000047    002 ST.UHL  000004      006 WORD5 = 000012
MSG2L = 000022         PUTFRE= ****** G       SR.ECH  000046    002 ST.XLT  000014      006 WORD6 = 000014
MSG2T   006724R        PUTQUT= ****** G       SR.ECL  000050    002 SUINDX= ****** G       WORD7 = 000016
MSG3    006746R        PUTSSQ= ****** G       SR.FIB  000012    002 SUST  = ****** G       WORD8 = 000020
MSG3L = 000020         QE.RD1= 000144         SR.GRE  000100    002 SU.DBU= 000004         WORD9 = 000022
MSG3T   006750R        QTABCR  002030R        SR.GRS  000072    002 SU.DON= 000006         WRAPUP  004366R
N       000002         QTABTR  001764R        SR.LEN  000122    002 SU.IDL= 000000         WRDVAL= 000024
NB.DEV= 000200         QTSRSP  001162R        SR.LIN  000066    002 SU.LOD= 000001         WRTSPL  006770R
NB.DIR= 000100         QT0RSP  000726R        SR.LIP  000062    002 SU.SRC= 000002         WSRCGO  001654R
NB.NAM= 000004         QT0TCL  001064R        SR.MON  000006    002 SU.SRR= 000005         WTSFOS  001412R
NB.SD1= 000400         QUQEMP  002214R        SR.NDC  000042    002 SU.XPD= 000003         WTSSRC  001566R
NB.SD2= 001000         QUQENE  000566R        SR.NDS  000036    002 SXPDIN  003712R        WTSXPD  001346R
NB.SNM= 000040         Q.FDSC  000004    007 SR.NIN  000030    002 SYSFLG= ****** G       XBATCH= 000013
NB.STP= 000020         Q.NOBK  000000    007 SR.NIP  000022    002 S.FATT= 000016         XDBLOA= 000004
NB.SVR= 000010         Q.NUHL  000002    007 SR.SDB  000032    002 S.FDB = 000140         XDBPRO= 000012
NB.TYP= 000002         Q.SIZE  000014    007 SR.SRC  000002    002 S.FNAM= 000006         XDMCIN= 000006
NB.VER= 000001         REPRCB  005142R        SR.SUN  000000    002 S.FNB = 000036         XFOSMR= 000007
NEXTST  000006R        REQQRY  006432R        SR.TWS  000056    002 S.FNBW= 000017         XGTSRE= 000014
NOTIME  000040R        RLSLOK= ****** G       SR.WSL  000052    002 S.FNTY= 000004         XHITSK= 000011
NXBCSH  004436R        RMBIGQ  001742R        SR.YR   000004    002 S.FTYP= 000002         XHLMER= 000002
NXTBIN  002300R        RPRHLM  002140R        SR.1IN  000024    002 S.HRL = 000240         XHOTSK= 000010
NXTENT  000174R        RPRQT0  002116R        SR.1IP  000016    002 S.NFEN= 000020         XMSCHE= 000000
NXTQRP  005210R        RQLRPR  005200R        SSQMAT  000242R        TAQRPR  002564R        XQTS  = 000003
N.BFAC= 000004         RQMDBR  003126R        SSRCIN  004344R        TIMAST  007152R        XQT0  = 000001
N.BHGH= 000006         RQQTRM  005006R        SSULOD  003670R        TOPNOD  000452R        XSULOA= 000005
N.BTCH= 000004         RUNSCH  005504R        SSUXDN  002644R        TRACE   006052R        $DDIV = ****** G
N.BUFB= 004000         RUNTSK  005322R        SS.FID  000002    004 TRCISO  006064R        $$$ARG= 000002
N.BUFW= 002000         RUNXTS  003316R        SS.FNB  000010    004 TRCLUN= 000004         $$$OST= 000020
N.DID = 000024         R.FIX = 000001         SS.FVR  000006    004 TRUE    002204R        $$$T2 = 000000
N.DVNM= 000032         R.SEQ = 000003         SS.LEN  000012    004 TSTDND  000332R        .CLOSE= ****** G
N.FID = 000000         R.VAR = 000002         SS.STT  000000    004 TULOUQ  002636R        .DLFNB= ****** G
N.FNAM= 000006         SALORP  002162R        STATDN  004402R        UPRBRS  00A664R        .OPFNB= ****** G
N.FOS = 000764         SCDBUD  004250R        STMDBU  004604R        WAITSG  001260R        .WAIT = ****** G
N.FTYP= 000014         SCHNAM  005540R        STRTSC  004150R        WAITSQ  000046R        .WRITE= ****** G
N.FVER= 000016         SETACT  003344R        STRTWO  006310R        WN    = 000000             ...GBL= 000000
N.NEXT= 000022         SEZLOK= ****** G       STTENT  000000RG       WN.NTP  000004      012 ...TPC= 000140
N.PKSZ= 000020         SFOSIN  004012R

. ABS.  000000  000
        007210  001
SRCOFF  000122  002
FDSCOF  000010  003
SUSOFF  000012  004
DHROFF  000012  005
STTOFF  000044  006
QSPLOF  000014  007
BSTOFF  000772  010
```

SYMBOL·TABLE·


```
FNOFFS  000044     011
WNODOF  000010     012
DNODOF  000010     013
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  7295 WORDS  (29 PAGES)
DYNAMIC MEMORY:  8084 WORDS  ( 31 PAGES)
ELAPSED TIME:  00:01:40
STT,STT/-SP=[20,1]P,M,STT,STTN,STTD,STTC,STTS,END
```

10- 2 MASTER COMPUTER COMMON

```
                                .TITLE  MCOM
                                .SBTTL  MASTER COMPUTER COMMON
000000                          .PSECT  MCOM
                        ;
                        ;
                                .MCALL  SETF$S,DECL$S
                                .MCALL  ALTP$S,MRKT$S,WTSE$S
                        ;
                        ; STATUS RECORDS
                        ;
000000                  SRECPT::
                        N=0
000000                  .REPT   N,SUNT
000002                  .IRP    Z,<\N>
                                .WORD   SREC'Z          ;ADDRESS OF DATA
                        .ENDR
                        N=N+1
                        .ENDR
                        ;
                        ;
                        N=0
000000                  .REPT   N,SUNT
000002                  .IRP    Z,<\N>
                        SREC'Z: .BLKB   SR.LEN          ;DATA FOR SEARCH UNIT
                        .ENDR
                        N=N+1
                        .ENDR
                        ;
                        ;
                        ; SEARCH UNIT STATUS TABLES
                        ;
000250                  SUINDX::
                        N=0
000000                  .REPT   N,SUNT
000002                  .IRP    Z,<\N>
                                .WORD   SUST'Z          ;ADDRESS OF DATA
                        .ENDR
                        N=N+1
                        .ENDR
                        ;
                        ;
000254  177777                  .WORD   -1             ;BATCH NUMBER BEING SEARCHED IN LOW BYTE
                                                       ; (-1 IF NONE)
000256                  SUST::
                        N=0
000000                  .REPT   N,SUNT
000002                  .IRP    Z,<\N>
                        SUST'Z: .WORD   0              ;SEARCH UNIT STATUS
                                .BLKW   3              ;FDSC OF SY0:[300,NJFOS.SPL
                                .WORD   FN.SF'Z
                        .ENDR
                        N=N+1
                        .ENDR
                        ;
                        ;
                        ; STATISTICS AREA
                        ;
```

```
                        .STATSS::
000302                  CHSTAT::.BLKB   ST.LEN   ;CURRENT HOUR
000346                  CDSTAT::.BLKB   ST.LEN   ;CURRENT DAY
000412                  CWSTAT::.BLKB   ST.LEN   ;CURRENT WEEK
000456                  LHSTAT::.BLKB   ST.LEN   ;LAST HOUR
000522                  LDSTAT::.BLKB   ST.LEN   ;LAST DAY
000566                  LWSTAT::.BLKB   ST.LEN   ;LAST WEEK
000632                  STATSE::
                        ;
                        ;
                        ; HRSTAT CONTROL WORDS
                        ;
000632  000000          HRSTFG::.WORD   0        ;FLAGS
000634                  LDAY::  .BLKW   1        ;LAST DAY
000636                  LHOUR:: .BLKW   1        ;LAST HOUR
                        ;
                        ; QUERY TRANSLATOR STATISTICS
                        ;
000640                  QTSTAT::.BLKW   32.
                        ;
```

```
                                ;
                                ; IN-CORE DHR CONTROL TABLES
                                ;
                                ; INDEX TABLE
                                ;
000740                          SUDHRI::                ;START OF TABLE
        000000                  N=0                     ;SU 0
        000002                  .REPT   N.SUNT
                                .IRP    Z,<\N>
                                        .WORD   S'Z'DHRC        ;DHR CONTROL TABLE ADDRESS
                                .ENDR
                                N=N+1
                                .ENDR
                                ;
                                ; CONTROL TABLES
                                ;
        000000                  N=0                     ;SU 0
        000000                  M=0                     ;BUFFER NUMBER
        000002                  .REPT   N.SUNT
                                .IRP    Z,<\N>
                                S'Z'DHRC: .WORD   0              ;CONTROL WORD
                                        .WORD   N.BUFB*M        ;FIRST BUFFER
                                        .WORD   N.BUFB*<M+1>    ;SECOND BUFFER
                                        .WORD   CF.DR'Z         ;GLOBAL FLAG
                                        .WORD   0               ;DMCIN SAVE AREA
                                .ENDR
                                N=N+1                   ;NEXT SU
                                M=M+2                   ;NEXT BUFFERS
                                .ENDR
                                ;
```

```
                                ;
                                ;
                                ; BATCH·STATUS·TABLE DATA·
                                ;
000770  000000                  SYSFLG::.WORD·  0       ;SYSTEM-WIDE·FLAGS·
                                                        ;BIT0 - TRACE·SCHEDULER·
000772· 000000                          .WORD·  0       ;NOT·USED·
                                ;
                                ; BATCH·STATUS·TABLE POINTERS·
                                ;
000774                          BSTPTR::
                                ;
                        000000  N=0
                        000004  .REPT·  N.BTCH·         ;NUMBER·OF·BATCH STATUS·TABLES·
                                .IRP·   Z,<\N>
                                ;
                                        .WORD·  BST'Z·  ;BST·ADDRESS·
                                ;
                                .ENDR·
                                N=N+2·
                                .ENDR·
                                ;
                                ;
                                ; BATCH·STATUS·TABLES·- BST0, BST1, ETC·
                                ;
                        000000  N=0
                        000004  .REPT·  N.BTCH·
                                .IRP·   Z,<\N>
                                ;
                                BST'Z:                  ;START OF·BST·
                                        .WORD·  -1      ;SSQ·INTERLOCK·
                                        .WORD·  0       ;EMPTY SSQ - COUNT·
                                        .WORD·  .+0     ;            FORWARD·POINTER·
                                        .WORD·  .-2·    ;            LAST·ENTRY·ON LIST·
                                        .WORD·  0       ;SCHEDULER'S·FDB ADDRESS·
                                        .BYTE·  0       ;SEND·PACKET·- MSCHED·COMMAND·SOURCE
                                        .BLKB·  1       ;              COMMAND·TYPE·
                                        .BLKW·  12.     ;              DATA·
                                        .WORD·  0       ;SCHEDULER·STATUS·
                                        .WORD·  0       ;CURRENT/NEXT·NODE·
                                        .BYTE·  CF.B'Z· ;SSQ·FLAG·
                                        .BYTE·  0       ;NOT·USED·
                                        .BYTE·  N       ;BATCH NUMBER·
                                        .BYTE·  0       ;BATCH STATE·
                                        .WORD·  0       ;BATCH STATE·STATUS·FLAGS·
                                        .WORD·  0       ;QUQ·ENTRY·PRESENT·FLAG·
                                        .BLKW·  7       ;UNLOADED·QUQ·ENTRY·
                                        .WORD·  0       ;BATCH XLATE·TIME·
                                        .WORD·  0       ;FSA·SIZE·
                                        .WORD·  0       ;FSB·SIZE·
                                        .WORD·  0       ;FSC·SIZE·
                                        .WORD·  0       ;QLS·SIZE·
                                        .WORD·  0.0     ;#·OF·UHL·DOC·
                                        .WORD·  0.0     ;#·OF·HRL·DOC·
                                        .WORD·  0       ;NUMBER·OF·UNUSED·BLOCK[illegible]
                                        .WORD·  0       ;START BLOCK·OF·PREVIOUS·HRL.MRG·(READ)
                                        .WORD·  0       ;START BLOCK·OF·NEW·HRL.MRG·(WRITE)
```

```
                .BLKW   N.SUNT  ;HRL SUB-FILES PER SEARCH UNIT
; FILE DESCRIPTORS FOR CONTROL TABLES
                .BLKW   3       ;FDSC FOR EMATRIX.EMA
                .WORD   FN.EMA
                .BLKW   3       ;FDSC FOR EMATRIX.EMB
                .WORD   FN.EMB
                .BLKW   3       ;FDSC FOR EMATRIX.EMC
                .WORD   FN.EMC
                .BLKW   3       ;FDSC FOR EMATRIX.QLS
                .WORD   FN.QLS
                .BLKW   3       ;FDSC FOR HRL.MRG
                .WORD   FN.MHR
                .BLKW   3       ;FDSC FOR TDCTA.FSA
                .WORD   FN.FSA
                .BLKW   3       ;FDSC FOR TDCTB.FSA
                .WORD   FN.FSB
                .BLKW   3       ;FDSC FOR TDCTC.FSA
                .WORD   FN.FSC
; INPUT QUERY ID MAPPING AREA
                .WORD   0       ;NUMBER OF QUERIES IN BATCH SO FAR
.REPT   N.QURY
                .WORD   0       ;MAP QID TO EQID
.ENDR
; QUERY SPOOL FILE AREA
.REPT   N.QURY
                .WORD   0       ;# OF QUERY BLOCKS
                .WORD   0       ;# OF UHL BLOCKS
                .BLKW   3       ;FDSC FOR QUERY.SPL
                .WORD   FN.QRY
.ENDR
;
; END OF BST
;
.ENDR
N=N+2
.ENDR
;
;
;
```

```
                                        ;
                                        ;
                                        ; BLDNFL, BLDEFL DATA STRUCTURES
                                        ;
                                        ; FILE NAME/TYPE INDEX
004754  005020'                        FNINDX: .WORD   FSANAM  ;FN.FSA
004756  005030'                                .WORD   FSBNAM  ;FN.FSB
004760  005040'                                .WORD   FSCNAM  ;FN.FSC
004762  005050'                                .WORD   QLSNAM  ;FN.QLS
004764  005060'                                .WORD   MHRNAM  ;FN.MHR
004766  005070'                                .WORD   EMANAM  ;FN.EMA
004770  005100'                                .WORD   EMBNAM  ;FN.EMB
004772  005110'                                .WORD   EMCNAM  ;FN.EMC
004774  005120'                                .WORD   QRYNAM  ;FN.QRY
004776  005130'                                .WORD   DBSNAM  ;FN.DBS
005000  005140'                                .WORD   MFONAM  ;FN.MFO
005002  005150'                                .WORD   DBRNAM  ;FN.DBR
005004  005160'                                .WORD   SF0NAM  ;FN.SF0
005006  005170'                                .WORD   SF1NAM  ;FN.SF1
005010  005200'                                .WORD   DB0NAM  ;FN.DB0
005012  005210'                                .WORD   DB1NAM  ;FN.DB1
005014  005220'                                .WORD   DHRNAM  ;FN.DHR
005016  005230'                                .WORD   SHDNAM  ;FN.SHD
                                        ;
                                        ; FILE NAME/TYPE
                                        ;
005020  076643  076450  000000  FSANAM: .RAD50  /TDCTA    FSA/  ;FN.FSA
005026  024171
005030  076643  076520  000000  FSBNAM: .RAD50  /TDCTB    FSA/  ;FN.FSB
005036  024171
005040  076643  076570  000000  FSCNAM: .RAD50  /TDCTC    FSA/  ;FN.FSC
005046  024171
005050  020511  077731  113000  QLSNAM: .RAD50  /EMATRIX  QLS/  ;FN.QLS
005056  066063
005060  032334  000000  000000  MHRNAM: .RAD50  /HRL      MRG/  ;FN.MHR
005066  052027
005070  020511  077731  113000  EMANAM: .RAD50  /EMATRIX  EMA/  ;FN.EMA
005076  020511
005100  020511  077731  113000  EMBNAM: .RAD50  /EMATRIX  EMB/  ;FN.EMB
005106  020512
005110  020511  077731  113000  EMCNAM: .RAD50  /EMATRIX  EMC/  ;FN.EMC
005116  020513
005120  066615  072150  000000  QRYNAM: .RAD50  /QUERY    SPL/  ;FN.QRY
005126  074514
005130  014545  062240  000000  DBSNAM: .RAD50  /DBUPD    SPL/  ;FN.DBS
005136  074514
005140  023753  000000  000000  MFONAM: .RAD50  /FQS      MRG/  ;FN.MFO
005146  052027
005150  014542  074264  000000  DBRNAM: .RAD50  /DBRSLT   SPL/  ;FN.DBR
005156  074514
005160  023753  000000  000000  SF0NAM: .RAD50  /FOS      SPL/  ;FN.SF0
005166  074514
005170  023753  000000  000000  SF1NAM: .RAD50  /FOS      SPL/  ;FN.SF1
005176  074514
005200  014545  062240  000000  DB0NAM: .RAD50  /DBUPD    BCH/  ;FN.DB0
005206  006400
005210  014545  062240  000000  DB1NAM: .RAD50  /DBUPD    BCH/  ;FN.DB1
```

```
     005216  006400
249  005220  015122  000000  000000  DHRNAM: .RAD50  /DHR      SPL/  ;FN.DHR
     005226  074514
250  005230  052073  031314  000000  SHDNAM: .RAD50  /MSCHED   SPL/  ;FN.SHD
     005236  074514
251                                  ;
252                                  ; FILE DIRECTORY FID INDEX
253                                  ;
254  005240  005312'                 FDINDX: .WORD   DIRP75  ;FN.FSA
255  005242  005312'                         .WORD   DIRP75  ;FN.FSB
256  005244  005312'                         .WORD   DIRP75  ;FN.FSC
257  005246  005312'                         .WORD   DIRP75  ;FN.QLS
258  005250  005312'                         .WORD   DIRP75  ;FN.MHR
259  005252  005312'                         .WORD   DIRP75  ;FN.EMA
260  005254  005312'                         .WORD   DIRP75  ;FN.EMB
261  005256  005312'                         .WORD   DIRP75  ;FN.EMC
262  005260  005304'                         .WORD   DIRP74  ;FN.QRY
263  005262  005304'                         .WORD   DIRP74  ;FN.DBS
264  005264  005304'                         .WORD   DIRP74  ;FN.MFO
265  005266  005304'                         .WORD   DIRP74  ;FN.DBR
266  005270  005320'                         .WORD   DIRP31  ;FN.SF0
267  005272  005326'                         .WORD   DIRP32  ;FN.SF1
268  005274  005320'                         .WORD   DIRP31  ;FN.DB0
269  005276  005326'                         .WORD   DIRP32  ;FN.DB1
270  005300  005304'                         .WORD   DIRP74  ;FN.DHR
271  005302  005304'                         .WORD   DIRP74  ;FN.SHD
272                                  ;
273                                  ; DIRECTORY FID'S
274                                  ;
275  005304                          DIRP74::.BLKW   3       ;SY0:[7,4]
276  005312                          DIRP75::.BLKW   3       ;SY0:[7,5]
277  005320                          DIRP31::.BLKW   3       ;SY0:[300,1]
278  005326                          DIRP32::.BLKW   3       ;SY0:[300,2]
279                                  ;
280                                  ; DEVICE NAME INDEX
281                                  ;
282  005334      123     131         DVINDX: .ASCII  /SY/    ;FN.FSA
283  005336      123     131                 .ASCII  /SY/    ;FN.FSB
284  005340      123     131                 .ASCII  /SY/    ;FN.FSC
285  005342      123     131                 .ASCII  /SY/    ;FN.QLS
286  005344      123     131                 .ASCII  /SY/    ;FN.MHR
287  005346      123     131                 .ASCII  /SY/    ;FN.EMA
288  005350      123     131                 .ASCII  /SY/    ;FN.EMB
289  005352      123     131                 .ASCII  /SY/    ;FN.EMC
290  005354      123     131                 .ASCII  /SY/    ;FN.QRY
291  005356      123     131                 .ASCII  /SY/    ;FN.DBS
292  005360      123     131                 .ASCII  /SY/    ;FN.MFO
293  005362      123     131                 .ASCII  /SY/    ;FN.DBR
294  005364      123     131                 .ASCII  /SY/    ;FN.SF0
295  005366      123     131                 .ASCII  /SY/    ;FN.SF1
296  005370      123     131                 .ASCII  /SY/    ;FN.DB0
297  005372      123     131                 .ASCII  /SY/    ;FN.DB1
298  005374      123     131                 .ASCII  /SY/    ;FN.DHR
299  005376      123     131                 .ASCII  /SY/    ;FN.SHD
300                                  ;
301                                  ; DEVICE UNIT INDEX
302                                  ;
```

```
303 005400  000000                UNINDX:: .WORD  0       ;FN.FSA
304 005402  000000                         .WORD  0       ;FN.FSB
305 005404  000000                         .WORD  0       ;FN.FSC
306 005406  000000                         .WORD  0       ;FN.QLS
307 005410  000000                         .WORD  0       ;FN.MHR
308 005412  000000                         .WORD  0       ;FN.EMA
309 005414  000000                         .WORD  0       ;FN.EMB
310 005416  000000                         .WORD  0       ;FN.EMC
311 005420  000000                         .WORD  0       ;FN.QRY
312 005422  000000                         .WORD  0       ;FN.DBS
313 005424  000000                         .WORD  0       ;FN.MFO
314 005426  000000                         .WORD  0       ;FN.DBR
315 005430  000000                         .WORD  0       ;FN.SF0
316 005432  000000                         .WORD  0       ;FN.SF1
317 005434  000000                         .WORD  0       ;FN.DB0
318 005436  000000                         .WORD  0       ;FN.DB1
319 005440  000000                         .WORD  0       ;FN.DHR
320 005442  000000                         .WORD  0       ;FN.SHD
321                               ;
```

```
                                        ;
                                        ;
                                        ; BUILD FILE NAME BLOCKS IN FDB
                                        ;
                                                .MCALL  FDOF$L,FCSBT$
005444                                          FDOF$L
005444                                          FCSBT$
                                        ;
                                        ; BLDNFL - BUILD FILE NAME BLOCK FOR NEW FILE
                                        ;
                                        ;   INPUT: R0 - FDB ADDRESS
                                        ;          R1 - FILE NUMBER (FN.XXX)
                                        ;  OUTPUT: ALL REGISTERS PRESERVED
                                        ;
005444  062700  000102                  BLDNFL::ADD     #F.FNB,R0       ;POINT TO FILE NAME BLOCK
005450  005060  000000                          CLR     N.FID(R0)       ;CLEAR FID
005454  005060  000002                          CLR     N.FID+2(R0)
005460  005060  000004                          CLR     N.FID+4(R0)
005464  010246                                  MOV     R2,-(SP)        ;SAVE R2
005466  016102  004754'                        MOV     FNINDX(R1),R2   ;FILE NAME/TYPE ADDRESS
005472  011260  000006                        MOV     (R2),N.FNAM(R0) ;FILE NAME/TYPE IN FDB
005476  016260  000002  000010                MOV     2(R2),N.FNAM+2(R0)
005504  016260  000004  000012                MOV     4(R2),N.FNAM+4(R0)
005512  016260  000006  000014                MOV     6(R2),N.FTYP(R0)
005520  005060  000016                        CLR     N.FVER(R0)      ;VERSION IS ZERO
005524  005060  000020                        CLR     N.STAT(R0)      ;CLEAR STATUS
005530  005060  000022                        CLR     N.NEXT(R0)      ;CLEAR WILD CARD WORD
005534  016102  005240'                        MOV     FDINDX(R1),R2   ;DIRECTORY FID ADDRESS
005540  011260  000024                        MOV     (R2),N.DID(R0)  ;DIRECTORY FID IN FDB
005544  016260  000002  000026                MOV     2(R2),N.DID+2(R0)
005552  016260  000004  000030                MOV     4(R2),N.DID+4(R0)
005560  016160  005334' 000032                MOV     DVINDX(R1),N.DVNM(R0)   ;DEVICE NAME
005566  016160  005400' 000034                MOV     UNINDX(R1),N.UNIT(R0)   ;DEVICE UNIT
005574  012602                                  MOV     (SP)+,R2        ;RESTORE R2
005576  162700  000102                        SUB     #F.FNB,R0       ;RESTORE R0
005602  000207                                  RTS     PC              ;RETURN
                                        ;
                                        ;
                                        ; BLDEFL - BUILD FILE NAME BLOCK FOR EXISTING FILE
                                        ;
                                        ;   INPUT: R0 - FDB ADDRESS
                                        ;          R1 - FILE DESCRIPTOR ADDRESS
                                        ;  OUTPUT: ALL REGISTERS PRESERVED
                                        ;
005604  010146                          BLDEFL::MOV     R1,-(SP)        ;SAVE FDSC ADDRESS
005606  016101  000006                          MOV     FD.FNB(R1),R1   ;GET FILE # FROM FDSC
005612                                          CALL    BLDNFL          ;BUILD SKELETON FILE NAME BLOCK
005616  012601                                  MOV     (SP)+,R1        ;RESTORE FDSC ADDRESS
005620  016160  000000  000102                MOV     FD.FID(R1),F.FNB+N.FID(R0)      ;ADD FID TO FILE NAME BLOCK
005626  016160  000002  000104                MOV     FD.FID+2(R1),F.FNB+N.FID+2(R0)
005634  016160  000004  000120                MOV     FD.FVR(R1),F.FNB+N.FVER(R0)     ;ADD VERSION
005642  000207                                  RTS     PC              ;RETURN
                                        ;
                                        ;
```

```
                                ;
                                ; PACKET POOL AND HEAD CELLS
                                ;
                                ;
                                ; FREE PACKET LIST HEAD CELLS
                                ;
005644  177777                          .WORD   -1              ;INTERLOCK
005646  000043                  NFREEP::.WORD   N.PKTS           ;COUNT
005650  005654'                  PKTFRE: .WORD   PKTSTT           ;FORWARD
005652  006714'                          .WORD   PKTEND          ;LAST
                                ;
                                ;
                                ; PACKET POOL
                                ;
005654                          PKTSTT:                         ;START OF POOL
        000042                  .REPT   N.PKTS-1
                                        .WORD   .+N.PKSZ        ;FORWARD LINK
                                        .BLKB   N.PKSZ-2
                                .ENDR
006714  005650'                 PKTEND: .WORD   PKTFRE          ;LAST PACKET - POINT TO HEAD CELL
006716                                  .BLKB   N.PKSZ-2
                                ;
                                ;
                                ; QUO HEAD CELL
006734  177777                          .WORD   -1              ;INTERLOCK
006736  000000                          .WORD   0               ;COUNT
006740  006740'                 QUOHED: .WORD   .+0             ;FORWARD
006742  006740'                          .WORD   .-2            ;LAST
                                ;
                                ;
```

```
                                ;
                                ;
                                ; PACKET HANDLING ROUTINES - ALL SAVE AND RESTORE REGISTERS
                                ;                           - ALL ARE RE-ENTRANT
                                ;                           - ALL "PUT" ROUTINES RANGE-CHECK PACKET ADDRESS
                                ;                           - PUTSSQ CHECKS FOR VALID COMMAND CODE
                                ;
                                ;
                                ; GET FREE PACKET FROM TOP OF FREE LIST
                                ;
                                ; INPUT - NONE
                                ; OUTPUT - R2 = ADDRESS OF PACKET
                                ;
006744                          GETFRE::SAVE    R3
006746  012703  005650'                MOV     #PKTFRE,R3      ;HEAD CELL ADDRESS
006752  005763  177776                 TST     -2(R3)          ;BRANCH IF NOT -
006756  001002                         BNE     1$              ;  EMPTY
006760                                 CALL    PKTERR
                                ;
006764                          1$:    CALL    GETTOP          ;DEQUEUE FROM TOP
006770                                 RESTOR  R3
006772  000207                         RTS     PC
                                ;
                                ;
                                ; PUT FREE PACKET ON BOTTOM OF FREE LIST
                                ;
                                ; INPUT - R2 = ADDRESS OF PACKET
                                ; OUTPUT - NONE
                                ;
006774                          PUTFRE::SAVE    R3
006776  012703  005650'                MOV     #PKTFRE,R3      ;HEAD CELL ADDRESS
007002                                 CALL    PUTBOT          ;QUEUE TO BOTTOM
007006                                 RESTOR  R3
007010  000207                         RTS     PC
                                ;
                                ;
                                ; GET TOP QUQ ENTRY
                                ;
                                ; INPUT - R5 = BST ADDRESS
                                ; OUTPUT - PACKET UNLOADED - B.QUQP = 0
                                ;                           B.CQUQ - 7 WORDS OF DATA
                                ;          NO PACKET TO UNLOAD - B.QUQP = 0
                                ;
007012  016765  177720  000056  GETQUQ::MOV     QUQHED-2,B.QUQP(R5) ;BRANCH IF
007020  001431                         BEQ     1$              ;  QUQ EMPTY
                                ;
007022                                 SAVE    R2              ;SAVE WORD REG
007024                                 SAVE    R3
007026  012703  006740'                MOV     #QUQHED,R3      ;ADDRESS OF QUQ HEAD
007032                                 CALL    GETTOP          ;GET TOP ENTRY
007036  062702  000002                 ADD     #2,R2           ;MOVE DATA TO BST
007042  062705  000060                 ADD     #B.CQUQ,R5
007046  012225                         MOV     (R2)+,(R5)+
007050  012225                         MOV     (R2)+,(R5)+
007052  012225                         MOV     (R2)+,(R5)+
007054  012225                         MOV     (R2)+,(R5)+
007056  012225                         MOV     (R2)+,(R5)+
```

```
 466 007060  012225                        MOV     (R2)+,(R5)+
 467 007062  011215                        MOV     (R2),(R5)
 468 007064  162702  000016                SUB     #14.,R2
 469 007070  162705  000074                SUB     #B,CQUQ+12.,R5  ;RESTORE BST ADDRESS
 470 007074                                CALL    PUTFRE          ;RETURN PACKET
 471 007100                                RESTOR  R3
 472 007102                                RESTOR  R2
 473                                ;
 474 007104  000207                 1$:    RTS     PC              ;RETURN
 475                                ;
 476                                ;
 477                                ; PUT QUQ ENTRY ON TOP/BOTTOM OF QUQ
 478                                ;
 479                                ; INPUT - R2 = QUQ PACKET ADDRESS
 480                                ; OUTPUT - NONE
 481                                ;
 482 007106                         PUTQUB::SAVE    R3
 483 007110  012703  006740'                MOV     #QUQHED,R3      ;QUQ HEAD CELL
 484 007114                                 CALL    PUTBOT          ;QUEUE TO BOTTOM
 485 007120  000405                         BR      PQUQCM
 486                                ;
 487 007122                         PUTQUT::SAVE    R3
 488 007124  012703  006740'                MOV     #QUQHED,R3      ;QUQ HEAD CELL
 489 007130                                 CALL    PUTTOP          ;QUEUE TO TOP
 490                                ;
 491 007134  022763  000001  177776 PQUQCM: CMP     #1,-2(R3)       ;HAS Q GONE NON-EMPTY?
 492 007142  001014                         BNE     1$              ;NO
 493                                ;PUT SSQ ENTRY IN FOR OPEN BATCH
 494 007144                                 SAVE    R2
 495 007146                                 CALL    GETFRE          ;GET A PACKET
 496 007152  112762  000000  000002        MOVB    #XMSCHED,2(R2)  ;SSQ NON EMPTY -
 497 007160  112762  000004  000003        MOVB    #4,3(R2)        ;  COMMAND
 498 007166                                 CALL    PUTSSQ          ;QUEUE TO OPEN-BATCH SCHEDULER
 499 007172                                 RESTOR  R2
 500 007174                         1$:     RESTOR  R3
 501 007176  000207                         RTS     PC
 502                                ;
 503                                ;
 504                                ; GET TOP ENTRY FROM THE SSQ
 505                                ;
 506                                ; INPUT - R3 = BST ADDRESS
 507                                ; OUTPUT - R2 = SSQ PACKET ADDRESS
 508                                ;                OR 0 IF SSQ EMPTY
 509                                ;
 510 007200  016302  000002         GETSSQ::MOV     B,SSQ-2(R3),R2  ;BRANCH IF -
 511 007204  001406                         BEQ     1$              ;  SSQ EMPTY
 512                                ;
 513 007206  062703  000004                 ADD     #B,SSQ,R3       ;POINT TO SSQ
 514 007212                                 CALL    GETTOP          ;DEQUEUE TOP ENTRY
 515 007216  162703  000004                 SUB     #B,SSQ,R3       ;RESTORE R3
 516 007222  000207                 1$:     RTS     PC
 517                                ;
 518                                ;
 519                                ; PUT ENTRY ON BOTTOM OF SSQ
 520                                ; BATCH NUMBER EITHER SPECIFIED IN PACKET (LOCATION IS COMMAND DEPENDANT)
 521                                ;   OR BATCH NUMBER IS IMPLIED BY COMMAND
 522                                ;
```

```
                                        ; INPUT - R2 = PACKET ADDRESS
                                        ; OUTPUT - NONE
                                        ;
007224                                  PUTSSQ::SAVE    R3
007226                                          CALL    MAPBNB          ;DETERMINE BATCH NUMBER
007232  103003                                  BCC     1$              ;GOT BATCH NUMBER
007234                                          CALL    PUTFRE          ;OTHERWISE RELEASE PACKET
007240  000417                                  BR      PSSQDN          ;DONE
                                        ;
                                        ;GOT A BATCH NUMBER IN R3
007242  016303  000774'                 1$:     MOV     BSTPTR(R3),R3   ;BST ADDRESS
007246  062703  000004                          ADD     #B.SSQ,R3       ;SSQ ADDRESS
007252                                          CALL    PUTBOT          ;PUT PACKET ON SSQ BOTTOM
007256  116303  000044                          MOVB    B.SSQF-B.SSQ(R3),R3 ;SCHEDULER EVENT FLAG
007262                                          SETF$S  R3              ;SET EVENT FLAG
007272                                          DECL$S
007300                                  PSSQDN: RESTOR  R3
007302  000207                                  RTS     PC
                                        ;
                                        ;
```

```
                                        ;
                                        ;
                                        ; GET BATCH NUMBER FROM PACKET
                                        ;
                                        ; INPUT - R2 = ADDRESS OF PACKET
                                        ; OUTPUT - R3 = BATCH NUMBER AND CARRY CLEAR IF SUCCESS
                                        ;              OR CARRY SET IF NO BATCH FOUND FOR COMMAND
                                        ;
007304                                  MAPBNB: CALL    RNGPKT          ;RANGE CHECK PACKET ADDRESS
007310                                          SAVE    R4
007312  012704  007406'                          MOV     #TBNMAP,R4      ;MAPPING TABLE
                                        ;
007316  012403                          MAPBLP: MOV     (R4)+,R3        ;COMMAND WORD
007320  100002                                  BPL     1$              ;NOT END OF TABLE
007322                                          CALL    PKTERR          ;INVALID COMMAND
007326  020362  000002                  1$:     CMP     R3,2(R2)        ;BRANCH IF COMMAND -
007332  001402                                  BEQ     FNDMAT          ;  MATCHES PACKET
007334  005724                                  TST     (R4)+           ;SKIP WORD IN TABLE
007336  000767                                  BR      MAPBLP          ;NEXT ENTRY
                                        ;FOUND COMMAND MATCH
007340  011404                          FNDMAT: MOV     (R4),R4         ;BATCH DISPLACEMENT?
007342  100404                                  BMI     IMPLID          ;NO - BATCH NUMBER IMPLIED
007344  060204                                  ADD     R2,R4           ;ADDRESS IN PACKET OF BATCH NUMBER
                                                                        ;(CARRY IS CLEAR)
007346  011403                                  MOV     (R4),R3         ;BATCH NUMBER
007350                                  MAPEXT: RESTOR  R4
007352  000207                                  RTS     PC
                                        ;
                                        ;BATCH NUMBER IS IMPLIED - R4= BATCH STATE
007354  012703  000003                  IMPLID: MOV     #N.BTCH-1,R3    ;HIGHEST BATCH -
007360  060303                                  ADD     R3,R3           ;  NUMBER
                                        ;
007362  016346  000774'                 IMPLOP: MOV     BSTPTR(R3),-(SP) ;BST ADDRESS
007366  062716  000053                          ADD     #B:STTE,(SP)    ;BATCH STATE ADDRESS
007372  120436                                  CMPB    R4,@(SP)+       ;BRANCH IF STATES -
007374  001765                                  BEQ     MAPEXT          ;  MATCH (CARRY IS CLEAR)
007376  162703  000002                          SUB     #2,R3           ;CONTINUE IF -
007402  103367                                  BCC     IMPLOP          ;  MORE BST'S
007404  000761                                  BR      MAPEXT          ;NO STATE MATCHES (CARRY IS SET)
                                        ;
                                        ;
                                        ;
                                        ; TABLE TO DETERMINE BATCH NUMBER LOCATION IN COMMAND PACKET
                                        ;
                                        ; FIRST WORD - COMMAND
                                        ; SECOND WORD - BIT 15=0 : DISPLACEMENT OF BN IN PACKET
                                        ;               BIT 15=1 : BATCH STATE
                                        ;
                                        ; TABLE TERMINATED BY -1
                                        ;
007406                                  TBNMAP:
007406     001      000                         .BYTE   XQT0,0
007410  000004                                  .WORD   4
007412     001      001                         .BYTE   XQT0,1
007414  000004                                  .WORD   4
007416     001      003                         .BYTE   XQT0,3
007420  000004                                  .WORD   4
```

```
601 007422    001     004             .BYTE   XQT0,4
602 007424  000004                    .WORD   4
603 007426    001     005             .BYTE   XQT0,5
604 007430  000004                    .WORD   4
605 007432    001     002             .BYTE   XQT0,2
606 007434  000004                    .WORD   4
607 007436    001     006             .BYTE   XQT0,6
608 007440  000004                    .WORD   4
609 007442    001     007             .BYTE   XQT0,7
610 007444  000004                    .WORD   4
611 007446    002     000             .BYTE   XHLMERG,0
612 007450  000004                    .WORD   4
613 007452    002     001             .BYTE   XHLMERG,1
614 007454  000004                    .WORD   4
615 007456    002     002             .BYTE   XHLMERG,2
616 007460  000004                    .WORD   4
617 007462    002     003             .BYTE   XHLMERG,3
618 007464  000004                    .WORD   4
619 007466    002     004             .BYTE   XHLMERG,4
620 007470  000004                    .WORD   4
621 007472    003     000             .BYTE   XQTS,0
622 007474  000004                    .WORD   4
623 007476    003     001             .BYTE   XQTS,1
624 007500  000004                    .WORD   4
625 007502    005     000             .BYTE   XSULOAD,0
626 007504  000004                    .WORD   4
627 007506    006     000             .BYTE   XDMCIN,0
628 007510  100003                    .WORD   BIT15+BS.SRC
629 007512    006     001             .BYTE   XDMCIN,1
630 007514  100003                    .WORD   BIT15+BS.SRC
631 007516    006     002             .BYTE   XDMCIN,2
632 007520  100004                    .WORD   BIT15+BS.DBU
633 007522    006     003             .BYTE   XDMCIN,3
634 007524  100004                    .WORD   BIT15+BS.DBU
635 007526    007     001             .BYTE   XFOSMRG,1
636 007530  000004                    .WORD   4
637 007532    012     000             .BYTE   XDBPROC,0
638 007534  000004                    .WORD   4
639 007536    013     000             .BYTE   XBATCH,0
640 007540  000004                    .WORD   4
641 007542    013     001             .BYTE   XBATCH,1
642 007544  100001                    .WORD   BIT15+BS.OPN
643 007546    013     002             .BYTE   XBATCH,2
644 007550  100001                    .WORD   BIT15+BS.OPN
645 007552    000     000             .BYTE   XMSCHED,0
646 007554  000004                    .WORD   4
647 007556    000     001             .BYTE   XMSCHED,1
648 007560  000004                    .WORD   4
649 007562    000     002             .BYTE   XMSCHED,2
650 007564  000004                    .WORD   4
651 007566    000     004             .BYTE   XMSCHED,4
652 007570  100001                    .WORD   BIT15+BS.OPN
653                             ;
654 007572  177777                    .WORD   -1
655                             ;
656 007574                            .BLKW   10.*2   ;SPARE ENTRIES
```

```
                                        ;
                                        ;
                                        ; GENERAL.QUEUE.MANIPULATION.ROUTINES.
                                        ;
                                        ; INPUT.- R3 =.HEAD.CELL.ADDRESS.
                                        ;         R2 =.PACKET.ADDRESS.IF."PUT"
                                        ; OUTPUT.- R2 =.PACKET.ADDRESS.IF."GET"
                                        ;                 (QUEUE.MUST.NOT.BE.EMPTY.ON.GET)
                                        ;
007644                                  GETTOP: CALL.   SEZLOK.         ;SIEZE.INTERLOCK.
007650  011302.                                 MOV.    (R3),R2.        ;PACKET.ADDRESS.
007652.011213                                   MOV.    (R2),(R3)       ;NEW.TOP.OF.LIST.
007654  005363  177776                         DEC.    -2(R3)          ;ONE.LESS.PACKET.
007660  001002.                                BNE.    DONEQ.          ;QUEUE.NOT.EMPTY.
007662.010363  000002                         MOV.    R3,2(R3)        ;LAST.POINTS.TO.FORWARD.LINK
007666                                  DONEQ:  CALL.   RLSLOK.         ;RELEASE.INTERLOCK
007672.000207                                   RTS.    PC.
                                        ;
                                        ;
007674                                  PUTBOT: CALL.   SEZLOK.         ;SIEZE.INTERLOCK.
007700                                          CALL.   RNGPKT.         ;RANGE.CHECK.PACKET.ADDRESS.
007704  010273  000002                         MOV.    R2,@2(R3)       ;CHAIN.NEW.PKT.TO.LIST.BOTTOM.
007710  010263  000002                         MOV.    R2,2(R3)        ;HEAD.CELL.POINTS.TO.NEW.BOTTOM.
007714  010312.                                MOV.    R3,(R2)         ;NEW.PKT.POINTS.TO.F.IN.HEAD.CELL.
007716  005263  177776                         INC.    -2(R3)          ;INC.COUNT.
007722.000761                                   BR      DONEQ.
                                        ;
                                        ;
007724                                  PUTTOP: CALL.   SEZLOK.         ;SIEZE.INTERLOCK.
007730                                          CALL.   RNGPKT.         ;RANGE.CHECK.PACKET.ADDRESS.
007734  011312.                                MOV.    (R3),(R2)       ;CHAIN.TOP.OF.LIST.TO.NEW.PKT.
007736  010213                                  MOV.    R2,(R3)         ;F.IN.HEAD.CELL.POINTS.TO.NEW.TOP.
007740  005263  177776                         INC.    -2(R3)          ;INC.COUNT.
007744  022763  000001  177776                 CMP.    #1,-2(R3)       ;QUEUE.GONE.NON-EMPTY?.
007752.001345                                   BNE.    DONEQ.          ;NO.
007754  010263  000002                         MOV.    R2,2(R3)        ;YES.- B.POINTS.TO.NEW.PKT.TOO.
007760  000742.                                BR      DONEQ.
                                        ;
                                        ;
                                        ;
                                        ; QUEUE.INTERLOCK.ROUTINES.
                                        ;
                                        ; INPUT.- R3 =.HEAD.CELL.ADDRESS.
                                        ; OUTPUT.- NONE.
                                        ;
007762.005263  177774                   SEZLOK::INC.    -4(R3)          ;SIEZE.INTERLOCK.
007766  001001                                  BNE.    LKFAIL.         ;DIDN'T.GET.IT.
007770  000207                                  RTS.    PC.             ;GOT.IT.
                                        ;
                                        ; FAILED.TO.GET.A.LOCK.ON.THE.QUEUE.- SOMEONE.ELSE.MUST.HAVE.IT.
007772.005363  177774                   LKFAIL: DEC.    -4(R3)          ;RELEASE.OUR.LOCK.
007776                                          ALTP$S. ,#4.            ;LOWER.OUR.PRIORITY.
010014                                          MRKT$S. #20.,#10.,#1    ;WAIT.A.WHILE.
010040                                          WTSE$S. #20.
010052.                                         ALTP$S.                 ;RAISE.PRIORIT[redacted]
010066  000735                                  BR      SEZLOK.         ;TRY.TO.SEIZE.LOCK.AGAIN.
                                        ;
```

```
    715 010070  005363  177774          RLSLOK::DEC     -4(R3)          ;RELEASE INTERLOCK
    716 010074  000207                          RTS     PC
    717                                 ;
    718                                 ;
    719                                 ;
    720                                 ; RANGE CHECK PACKET ADDRESS
    721                                 ;
    722                                 ; INPUT - R2 = ADDRESS OF PACKET
    723                                 ; OUTPUT - GOOD: RETURN
    724                                 ;          BAD: CRASH
    725                                 ;
    726 010076  020227  005654'         RNGPKT: CMP     R2,#PKTSTT      ;LOWEST PACKET
    727 010102  103002                          BHIS    1$
    728 010104                                  CALL    PKTERR
    729 010110  020227  006714'         1$:     CMP     R2,#PKTEND      ;HIGHEST PACKET
    730 010114  101402                          BLOS    2$
    731 010116                                  CALL    PKTERR
    732 010122  000207                  2$:     RTS     PC
    733                                 ;
    734                                 ;
    735 010124  000167  177775          PKTERR: JMP     .+1             ;PACKET PROBLEM - CRASH
    736                                 ;
    737                                 ;
    738 010130                          MCOMSZ:
    739                                 ;
    740         000001                          .END
```

```
BITVAL= 000000          B.HRLR  000122          010 FA.EXT= 000004          FN.MHR  000010      011 F.SPUN= 000074
BIT0  = 000001          B.HRLW  000124          010 FA.NSP= 000100          FN.NMB  000044      011 F.STBK= 000036
BIT1  = 000002          B.NMBR  000052          010 FA.POS= 010000          FN.QLS  000006      011 F.UNIT= 000136
BIT10 = 002000          B.NQRY  000232          010 FA.RD = 000001          FN.QRY  000020      011 F.URBD= 000020
BIT11 = 004000          B.QLSZ  000106          010 FA.RWD= 004000          FN.SF0  000030      011 F.VBN = 000064
BIT12 = 010000          B.QMAP  000234          010 FA.SEQ= 040000          FN.SF1  000032      011 F.VBSZ= 000060
BIT13 = 020000          B.QSPL  000316          010 FA.SHR= 000040          FN.SHD  000042      011 GETFRE  006744RG    014
BIT14 = 040000          B.QTTM  000076          010 FA.TMP= 000020          FO.APD= 000106          GETQUQ  007012RG    014
BIT15 = 100000          B.QUQP  000056          010 FA.WCK= 020000          FO.MFY= 000002          GETSSQ  007200RG    014
BIT2  = 000004          B.SFDB  000010          010 FA.WRT= 000002          FO.RD = 000001          GETTOP  007644R     014
BIT3  = 000010          B.SIZE  000772          010 FDINDX  005240R     014 FO.UPD= 000006          HRSTFG  000632RG    014
BIT4  = 000020          B.SNDP  000012          010 FD.BLK= 000010          FO.WRT= 000016          IMPLID  007354R     014
BIT5  = 000040          B.SSQ   000004          010 FD.CCL= 000002          FSANAM  005020R     014 IMPLOP  007362R     014
BIT6  = 000100          B.SSQF  000050          010 FD.COM= 020000          FSBNAM  005030R     014 LDAY    000634RG    014
BIT7  = 000200          B.STAT  000044          010 FD.CR = 000002          FSCNAM  005040R     014 LDSTAT  000522RG    014
BIT8  = 000400          B.STTE  000053          010 FD.DIR= 000010          F.ACTL= 000076          LHOUR   000636RG    014
BIT9  = 001000          B.UDOC  000110          010 FD.FID  000000      003 F.ALOC= 000040          LHSTAT  000456RG    014
BLDEFL  005604RG    014 CDSTAT  000346RG      014 FD.FNB  000006      003 F.BBFS= 000062          LKFAIL  007772R     014
BLDNFL  005444RG    014 CF.B0 = 000070              FD.FTN= 000001          F.BDB = 000070          LWSTAT  000566RG    014
BSTPTR  000774RG    014 CF.B2 = 000067              FD.FVR  000004      003 F.BGBC= 000057          M       = 000004
BST0    001004R     014 CF.B4 = 000066              FD.FI1= 040000          F.BKDN= 000026          MAPBLP  007316R     014
BST2    001776R     014 CF.B6 = 000065              FD.INS= 000010          F.BKDS= 000020          MAPBNB  007304R     014
BST4    002770R     014 CF.DR0= 000064              FD.ISP= 002000          F.BKEF= 000050          MAPEXT  007350R     014
BST6    003762R     014 CF.DR1= 000063              FD.LEN  000010      003 F.BKP1= 000051          MCOMSZ  010130R     014
BS.CLS= 000002          CHSTAT  000302RG        014 FD.MNT= 100000          F.BKST= 000024          MFONAM  005140R     014
BS.DBU= 000004          CH.AND= 000001              FD.OSP= 004000          F.BKVB= 000064          MHRNAM  005060R     014
BS.INA= 000000          CWSTAT  000412RG        014 FD.PLC= 000004          F.CHR = 000075          N       = 000010
BS.OPN= 000001          DBRNAM  005150R             FD.PRN= 000004          F.CNTG= 000034          NB.DEV= 000200
BS.SRC= 000003          DBSLEN= 000116              FD.PSE= 010000          F.DFNB= 000046          NB.DIR= 000100
BYTE0 = 000000          DBSNAM  005130R         014 FD.RAH= 000001          F.DSPT= 000044          NB.NAM= 000004
BYTE1 = 000001          DB0NAM  005200R         014 FD.RAN= 000002          F.DVNM= 000134          NB.SD1= 000400
BYTE2 = 000002          DB1NAM  005210R         014 FD.REC= 000001          F.EFBK= 000010          NB.SD2= 001000
BYTE3 = 000003          DHRNAM  005220R         014 FD.RWM= 000001          F.EFN = 000050          NB.SNM= 000040
BYTE4 = 000004          DH.BF0  000002          005 FD.SDI= 000020          F.EOBB= 000032          NB.STP= 000020
BYTE5 = 000005          DH.BF1  000004          005 FD.SQD= 000040          F.ERR = 000052          NB.SVR= 000010
BYTE6 = 000006          DH.CTL  000000          005 FD.TTY= 000004          F.FACC= 000043          NB.TYP= 000002
BYTE7 = 000007          DH.DMC  000010          005 FD.WBH= 000002          F.FFBY= 000014          NB.VER= 000001
BYTE8 = 000010          DH.FLG  000006          005 FF.CHR= 000005          F.FNAM= 000110          NFREEP  005646RG    014
BYTE9 = 000011          DIRP31  005320RG        014 FF.NV = 000003          F.FNB = 000102          N.BFAC= 000004
BYTVAL= 000012          DIRP32  005326RG        014 FF.POE= 000002          F.FTYP= 000116          N.BHGH= 000006
B.BSTA  000054      010 DIRP74  005304RG        014 FF.RWD= 000001          F.FVER= 000120          N.BTCH= 000004
B.CNTX  000046      010 DIRP75  005312RG        014 FF.RWF= 000006          F.HIBK= 000004          N.BUFB= 004000
B.COUQ  000060      010 DN.DCK  000000          013 FF.SPC= 000004          F.LUN = 000042          N.BUFW= 002000
B.FEMA  000132      010 DN.NTP  000004          013 FNDMAT  007340R     014 F.MBCT= 000054          N.DID = 000024
B.FEMB  000142      010 DN.NXT  000006          013 FNINDX  004754R     014 F.MBC1= 000055          N.DVNM= 000032
B.FEMC  000152      010 DN.ROT  000002          013 FN.DBR  000026      011 F.MBFG= 000056          N.FID = 000000
B.FFSA  000202      010 DN.SIZ  000010          013 FN.DBS  000022      011 F.NRBD= 000024          N.FNAM= 000006
B.FFSB  000212      010 DONEQ   007666R         014 FN.DHR  000040      011 F.NREC= 000030          N.FOS = 000764
B.FFSC  000222      010 DVINDX  005334R         014 FN.EMA  000012      011 F.OVBS= 000030          N.FTYP= 000014
B.FMHR  000172      010 EMANAM  005070R         014 FN.EMB  000014      011 F.RACC= 000016          N.FVER= 000016
B.FQLS  000162      010 EMBNAM  005100R         014 FN.EMC  000016      011 F.RATT= 000001          N.NEXT= 000022
B.FSAZ  000100      010 EMCNAM  005110R         014 FN.FSA  000000      011 F.RCNM= 000034          N.PKSZ= 000020
B.FSBZ  000102      010 FA.APD= 000100              FN.FSB  000002      011 F.RCTL= 000017          N.PKTS= 000043
B.FSCZ  000104      010 FA.CRE= 000010              FN.FSC  000004      011 F.RSIZ= 000002          N.QUPV= 000031
B.HBLK  000120      010 FA.DLK= 001000              FN.LGQ  000034      011 F.RTYP= 000000          N.STAT= 000020
B.HDOC  000114      010 FA.ENA= 100000              FN.LGU  000036      011 F.SEQN= 000100          N.SUNT= 000002
B.HRLP  000126      010 FA.EXL= 002000              FN.MFQ  000024      011 F.SPDV= 000072          N.UNIT= 000034
```

```
MCOM  M1110  27-MAR-80 14:06  PAGE 18-3
SYMBOL TABLE

PKTEND  006714R     014 SF1NAM  005170R     014 SR.WSL  000052      002 SU.DBU=  000004          WORD0 =  000000
PKTERR  010124R     014 SHDNAM  005230R     014 SR.YR   000004      002 SU.DON=  000006          WORD1 =  000002
PKTFRE  005650R     014 SRECPT  000000RG    014 SR.1IN  000024      002 SU.IDL=  000000          WORD2 =  000004
PKTSTT  005654R     014 SREC0   000004R     014 SR.1IP  000016      002 SU.LOD=  000001          WORD3 =  000006
PQUQCM  007134R     014 SREC1   000126R     014 SS.FID  000002      004 SU.SRC=  000002          WORD4 =  000010
PSSQDN  007300R     014 SR.ARE  000114      002 SS.FNB  000010      004 SU.SRR=  000005          WORD5 =  000012
PUTBOT  007674R     014 SR.ARS  000106      002 SS.FVR  000006      004 SU.XPD=  000003          WORD6 =  000014
PUTFRE  006774RG    014 SR.DAY  000010      002 SS.LEN  000012      004 SYSFLG  000770RG    014 WORD7 =  000016
PUTQUB  007106RG    014 SR.DLT  000014      002 SS.STT  000000      004 S.FATT=  000016          WORD8 =  000020
PUTQUT  007122RG    014 SR.ECB  000047      002 STATSE  000632RG    014 S.FDB =  000140          WORD9 =  000022
PUTSSQ  007224RG    014 SR.ECH  000046      002 STATSS  000302RG    014 S.FNAM=  000006          WRDVAL=  000024
PUTTOP  007724R     014 SR.ECL  000050      002 ST.ASZ  000020      006 S.FNB =  000036          XBATCH=  000013
QE.RO1=  000144         SR.FIB  000012      002 ST.BSZ  000024      006 S.FNBW=  000017          XDBLOA=  000004
QLSNAM  005050R     014 SR.GRE  000100      002 ST.BTC  000000      006 S.FNTY=  000004          XDBPRO=  000012
QRYNAM  005120R     014 SR.GRS  000072      002 ST.CSZ  000030      006 S.FTYP=  000002          XDMCIN=  000006
QTSTAT  000540RG    014 SR.LEN  000122      002 ST.HRL  000010      006 S.HRL =  000240          XFOSMR=  000007
QUQHED  006740R     014 SR.LIN  000066      002 ST.LEN  000044      006 S.NFEN=  000020          XGTSRE=  000014
Q.FDSC  000004      007 SR.LIP  000062      002 ST.QRY  000002      006 S0DHRC  000744R     014 XHITSK=  000011
Q.NOBK  000000      007 SR.MON  000006      002 ST.QSZ  000034      006 S1DHRC  000756R     014 XHLMER=  000002
Q.NUHL  000002      007 SR.NDC  000042      002 ST.SCH  000040      006 TBNMAP  007406R     014 XHQTSK=  000010
Q.SIZE  000014      007 SR.NDS  000036      002 ST.UHL  000004      006 UNINDX  005400R     014 XMSCHE=  000000
RLSLOK  010070RG    014 SR.NIN  000030      002 ST.XLT  000014      006 WN.NTP  000004     012 XSTS  =  000003
RNGPKT  010076R     014 SR.NIP  000022      002 SUDHRI  000740RG    014 WN.NXT  000006     012 XQTB  =  000001
R.FIX =  000001         SR.SDB  000032      002 SUINDX  000250RG    014 WN.ROT  000002     012 XSULOA=  000005
R.SEQ =  000003         SR.SRC  000002      002 SUST    000256RG    014 WN.SIZ  000010     012 $$$ARG=  000004
R.VAR =  000002         SR.SUN  000000      002 SUST0   000256R     014 WN.SRC  000000     012 ...GBL=  000000
SEZLOK  007762RG    014 SR.TWS  000056      002 SUST1   000270R     014 WN.TYP  000001     012 ...TPC=  000140
SF0NAM  005160R     014

. ABS.  000000  000
        000000  001
SRCOFF  000122  002
FDSCOF  000010  003
SUSOFF  000012  004
DHROFF  000012  005
STTOFF  000044  006
QSPLOF  000014  007
BSTOFF  000772  010
FNOFFS  000044  011
WNODOF  000010  012
DNODOF  000010  013
MCOM    010130  014
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  5944 WORDS  (24 PAGES)
DYNAMIC MEMORY:  7028 WORDS  ( 27 PAGES)
ELAPSED TIME:  00:00:57
MCOM,MCOM/-SP=P,M,MCOM
```

```
                                        .TITLE  DHRCOM
                                        .SBTTL  DHR BUFFER COMMON
000000                                  .PSECT  DHRCOM
                                ;
                                ;
000000                          DHRCOM::
                                ;
                                ; GENERATE DHR BUFFERS
                                ;
000000                          N=0             ;SU 0
000002                          .REPT   N.SUNT
                                        .BLKW   N.BUFW  ;BUFFER 0
                                        .BLKW   N.BUFW  ;BUFFER 1
                                N=N+1           ;NEXT SU
                                .ENDR
                                ;
                                ;
020000                          DHRSZ:
                                ;
000001                                  .END
```

DHRCOM. MA[redacted] M1110 27-MAR-80 14:07
TABLE OF CONTENTS

10- 2 DHR BUFFER COMMON

```
BITVAL=000000          B.FMHR  000172      010 DN.NXT  000006      013 Q.NUHL  000002      007 ST.SCH  000040      006
BIT0  =000001          B.FQLS  000162      010 DN.ROT  000002      013 Q.SIZE  000014      007 ST.UHL  000004      006
BIT1  =000002          B.FSAZ  000100      010 DN.SIZ  000010      013 SR.ARE  000114      002 ST.XLT  000014      006
BIT10 =002000          B.FSBZ  000102      010 FD.FID  000000      003 SR.ARS  000106      002 SU.DBU=000004
BIT11 =004000          B.FSCZ  000104      010 FD.FNB  000006      003 SR.DAY  000010      002 SU.DON=000006
BIT12 =010000          B.HBLK  000120      010 FD.FVR  000004      003 SR.DLT  000014      002 SU.IDL=000000
BIT13 =020000          B.HDOC  000114      010 FD.LEN  000010      003 SR.ECB  000047      002 SU.LOD=000001
BIT14 =040000          B.HRLP  000126      010 FN.DBR  000026      011 SR.ECH  000046      002 SU.SRC=000002
BIT15 =100000          B.HRLR  000122      010 FN.DBS  000022      011 SR.ECL  000050      002 SU.SRR=000005
BIT2  =000004          B.HRLW  000124      010 FN.DHR  000040      011 SR.FIB  000012      002 SU.XPD=000003
BIT3  =000010          B.NMBR  000052      010 FN.EMA  000012      011 SR.GRE  000100      002 S.HRL =000240
BIT4  =000020          B.NQRY  000232      010 FN.EMB  000014      011 SR.GRS  000072      002 WN.NTP  000004      012
BIT5  =000040          B.QLSZ  000106      010 FN.EMC  000016      011 SR.LEN  000122      002 WN.NXT  000006      012
BIT6  =000100          B.QMAP  000234      010 FN.FSA  000000      011 SR.LIN  000066      002 WN.ROT  000002      012
BIT7  =000200          B.QSPL  000316      010 FN.FSB  000002      011 SR.LIP  000062      002 WN.SIZ  000010      012
BIT8  =000400          B.QTTM  000076      010 FN.FSC  000004      011 SR.MON  000006      002 WN.SRC  000000      012
BIT9  =001000          B.QUQP  000056      010 FN.LGQ  000034      011 SR.NDC  000042      002 WN.TYP  000001      012
BS.CLS=000002          B.SFDB  000010      010 FN.LGU  000036      011 SR.NDS  000036      002 WORD0 =000000
BS.DBU=000004          B.SIZE  000772      010 FN.MFO  000024      011 SR.NIN  000030      002 WORD1 =000002
BS.INA=000000          B.SNDP  000012      010 FN.MHR  000010      011 SR.NIP  000022      002 WORD2 =000004
BS.OPN=000001          B.SSQ   000004      010 FN.NMB  000044      011 SR.SDB  000032      002 WORD3 =000006
BS.SRC=000003          B.SSQF  000050      010 FN.QLS  000006      011 SR.SRC  000002      002 WORD4 =000010
BYTE0 =000000          B.STAT  000044      010 FN.QRY  000020      011 SR.SUN  000000      002 WORD5 =000012
BYTE1 =000001          B.STTE  000053      010 FN.SF0  000030      011 SR.TWS  000056      002 WORD6 =000014
BYTE2 =000002          B.UDOC  000110      010 FN.SF1  000032      011 SR.WSL  000052      002 WORD7 =000016
BYTE3 =000003          CF.B0 =000070           FN.SHD  000042      011 SR.YR   000004      002 WORD8 =000020
BYTE4 =000004          CF.B2 =000067           M       =000062         SR.1IN  000024      002 WORD9 =000022
BYTE5 =000005          CF.B4 =000066           N       =000002         SR.1IP  000016      002 WRDVAL=000024
BYTE6 =000006          CF.B6 =000065           N.BFAC=000004           SS.FID  000002      004 XBATCH=000013
BYTE7 =000007          CF.DR0=000064           N.BHGH=000006           SS.FNB  000010      004 XDBLOA=000004
BYTE8 =000010          CF.DR1=000063           N.BTCH=000004           SS.FVR  000006      004 XDBPRO=000012
BYTE9 =000011          DBSLEN=000116           N.BUFB=004000           SS.LEN  000012      004 XDMCIN=000006
BYTVAL=000012          DHRCOM  000000RG    014 N.BUFW=002000           SS.STT  000000      004 XFOSMR=000007
B.BSTA  000054      010 DHRSZ   020000R    014 N.FOS =000764           ST.ASZ  000020      006 XGTSRE=000014
B.CNTX  000046      010 DH.BF0  000002      005 N.PKSZ=000020           ST.BSZ  000024      006 XHITSK=000011
B.COUQ  000060      010 DH.BF1  000004      005 N.PKTS=000043           ST.BTC  000000      006 XHLMER=000002
B.FEMA  000132      010 DH.CTL  000000      005 N.QURY=000031           ST.CSZ  000030      006 XHOTSK=000010
B.FEMB  000142      010 DH.DMC  000010      005 N.SUNT=000002           ST.HRL  000010      006 XMSCHE=000000
B.FEMC  000152      010 DH.FLG  000006      005 QE.RQ1=000144           ST.LEN  000044      006 XQTS  =000003
B.FFSA  000202      010 DN.DCK  000000      013 Q.FDSC  000004      007 ST.QRY  000002      006 XQT0  =000001
B.FFSB  000212      010 DN.NTP  000004      013 Q.NOBK  000000      007 ST.QSZ  000034      006 XSULOA=000005
B.FFSC  000222      010

. ABS.  000000      000
        000000      001
SRCOFF  000122      002
FDSCOF  000010      003
SUSOFF  000012      004
DHROFF  000012      005
STTOFF  000044      006
QSPLOF  000014      007
BSTOFF  000772      010
FNOFFS  000044      011
WNODOF  000010      012
DNODOF  000010      013
DHRCOM  020000      014
ERRORS DETECTED:  0
```

```
VIRTUAL MEMORY USED:  2031 WORDS  ( 8 PAGES)
DYNAMIC MEMORY:  2804 WORDS  ( 10 PAGES)
ELAPSED TIME:  00:00:13
DHRCOM,DHRCOM/-SP=P,M,DHRCOM
```

```
.MAIN.  MACRO M1110  27-MAR-80 13:54  PAGE 1

2                                       .NLIST ME
```

```
                        ;
                        ; P:MAC - SYSTEM-WIDE PREFIX FILE
                        ;
                        ;
                        ;
000012                          .RADIX  10.
                        ;
                        ; BIT DEFINITIONS
                        ;
                        .MACRO BITDEF  X
                        BIT'X  = BITVAL                 ; SYMBOL SUBSTITUTE = BIT POSITION
                        .NLIST
                        N = N+1
                        BITVAL = BITVAL*2               ; SINGLE BIT LEFT SHIFT ONE
                        .LIST
                        .ENDM  BITDEF
000000                  N = 0                   ; BIT '0'
000001                  BITVAL = 1                      ; BIT0 IS ONE
000020                  .REPT  16.
                                BITDEF  \N              ; MACRO CALL, PASS CHARACTER
                        .ENDM
                        ;
                        ;WORD ADDRESS OFFSET DEFINITIONS
                        ;
                        .MACRO WORDOFF X
                        .NLIST
                        WORD'X = WRDVAL                 ; SYMBOL SUBSTITUTE = OCTAL VALUE
                        N = N + 1                       ; INCREMENT 'X'
                        WRDVAL = WRDVAL + 2
                        .LIST
                        .ENDM  WORDOFF
000000                  N = 0
000000                  WRDVAL = 0
000012                  .REPT  10.
                        .NLIST
                                WORDOFF \N              ; MACRO CALL, PASS CHARACTER
                        .LIST
                        .ENDM
                        ;
                        ; BYTE ADDRESS OFFSET DEFINITIONS
                        ;
                        .MACRO BYTEOFF X
                        .NLIST
                        BYTE'X = BYTVAL                 ; SYMBOL SUBSTITUTE = OCTAL VALUE
                        N = N+1                         ; INCREMENT 'X'
                        BYTVAL = BYTVAL+1
                        .LIST
                        .ENDM  BYTEOFF
000000                  N = 0
000000                  BYTVAL = 0
000012                  .REPT  10.
                        .NLIST
                                BYTEOFF \N              ; MACRO CALL, PASS CHARACTER
                        .LIST
                        .ENDM
000010                          .RADIX  8.
```

```
                        .MCALL CALL
                       ;
                       ; RETURN FROM SUBROUTINE - PC LINKAGE
                       ;
                       .MACRO  RTN
                               RTS     PC
                       .ENDM   RTN
                       ;
                       ;EXIT A SUBROUTINE
                        .MACRO EXIT    SUBR
                               RTS     PC
                       .ENDM   EXIT
                       ;
                       ; BRANCH ON ANY TESTED BIT ON -- USED AFTER BIT TEST (BIT OR BITB)
                        .MACRO BON     LOC
                        .NLIST
                               BNE     LOC             ; BRANCH IF BIT(S) SET
                        .LIST
                        .ENDM
                       ;
                       ; BRANCH ON ALL TESTED BIT(S) OFF -- USED AFTER BIT TEST (BIT OR BITB)
                        .MACRO BOFF    LOC
                        .NLIST
                               BEQ     LOC             ; BRANCH IF BIT(S) NOT SET
                        .LIST
                        .ENDM
                       ;
                        .MACRO SAVE    A1,A2,A3,A4,A5,A6
                        .IF    NB      <A1>
                               MOV     A1,-(SP)
                        .ENDC
                        .IF    NB      <A2>
                               MOV     A2,-(SP)
                        .ENDC
                        .IF    NB      <A3>
                               MOV     A3,-(SP)
                        .ENDC
                        .IF    NB      <A4>
                               MOV     A4,-(SP)
                        .ENDC
                        .IF    NB      <A5>
                               MOV     A5,-(SP)
                        .ENDC
                        .IF    NB      <A6>
                               MOV     A6,-(SP)
                        .ENDC
                        .ENDM  SAVE
                        .MACRO RESTOR  A1,A2,A3,A4,A5,A6
                        .IF    NB      <A6>
                               MOV     (SP)+,A6
                        .ENDC
                        .IF    NB      <A5>
                               MOV     (SP)+,A5
                        .ENDC
                        .IF    NB      <A4>
                               MOV     (SP)+,A4
                        .ENDC
```

```
                        .IF     NB      <A3>
                                MOV     (SP)+,A3
                        .ENDC
                        .IF     NB      <A2>
                                MOV     (SP)+,A2
                        .ENDC
                        .IF     NB      <A1>
                                MOV     (SP)+,A1
                        .ENDC
                        .ENDM   RESTOR
                        ;
                        ;
                        ;
                        ;       MESSAGE OUTPUT MACRO
                        ;
                        .MACRO  MOUT$S  MSG,PAR
                                MOV     MSG,-(SP)               ;PUSH ADDRESS OF ASCIZ STRING
                        .IF     NB      <PAR>
                                MOV     PAR,-(SP)               ;PUSH ADDRESS OF ARGUMENT BLOCK
                        .ENDC
                        .IF     B       <PAR>
                                CLR     -(SP)
                        .ENDC
                                JSR     PC,MSGOUT               ;EDIT OUTPUT MSG STRING AND PRINT IT
                                ADD     #4,SP                   ;RESTORE STACK POINTER
                        .ENDM
```

```
                                        ;
                                        ; SYSTEM EQUATES
                                        ;
          000004                        N.BTCH=4                        ;MAX NUMBER OF ACTIVE BATCHES
          000006                        N.BHGH=<N.BTCH-1>*2             ;HIGH BATCH NUMBER
          000031                        N.QURY=25.                      ;MAX # OF QUERIES IN A BATCH
          000764                        N.FOS=500.                      ;MAX # OF FOS ENTRIES (DOUBLE WORDS)
          002000                        N.BUFW=1024.                    ;SIZE OF SYSTEM BUFFERS (WORDS)
          004000                        N.BUFB=N.BUFW*2                 ;SIZE OF SYSTEM BUFFERS (BYTES)
          000004                        N.BFAC=N.BUFW/256.              ;# OF SECTORS IN SYSTEM BLOCK
          000002                        N.SUNT=2                        ;NUMBER OF SEARCH UNITS
                                        ;
                                        ;
                                        ; STATUS RECORD OFFSETS -SREC; SREC0; SREC1; ETC
                                        ;
 000000                                         .PSECT  SRCOFF,ABS
 000000                                 SR.SUN: .BLKW   1               ;SEARCH UNIT NUMBER
 000002                                 SR.SRC: .BLKW   1               ;SEARCH TIME OF LAST SEARCH (SEC)
                                        ; BEGINNING OF DATA BASE STATUS AREA
 000004                                 SR.YR:  .BLKW   1               ;YEAR
 000006                                 SR.MON: .BLKW   1               ;MONTH
 000010                                 SR.DAY: .BLKW   1               ;DAY OF DISK INIT
 000012                                 SR.FIB: .BLKW   1               ;FILE IDENTIFICATION BLOCK
 000014                                 SR.DLT: .BLKW   1               ;DELTA FOR INDEX SECTORS REPRESENTED
 000016                                 SR.1IP: .BLKW   2               ;ADDRESS OF FIRST IPR
 000022                                 SR.NIP: .BLKW   1               ;NUMBER OF IPR SECTORS ON DISK
 000024                                 SR.1IN: .BLKW   2               ;ADDRESS OF FIRST INDEX RECORD
 000030                                 SR.NIN: .BLKW   1               ;NUMBER OF INDEX RECORDS ALLOCATED
 000032                                 SR.SDB: .BLKW   2               ;ADDRESS OF START OF DATA BASE
 000036                                 SR.NDS: .BLKW   2               ;NUMBER OF DOCUMENTS AT INIT
 000042                                 SR.NDC: .BLKW   2               ;CURRENT NUMBER OF DOCUMENTS
 000046                                 SR.ECH: .BLKB   1               ;HIGH ORDER ADDRESS OF EOC
 000047                                 SR.ECB: .BLKB   1               ;BYTE INDEX OF EOC
 000050                                 SR.ECL: .BLKW   1               ;LOW ORDER ADDRESS OF EOC
 000052                                 SR.WSL: .BLKW   2               ;WHITE SPACE AFTER EOC
 000056                                 SR.TWS: .BLKW   2               ;TOTAL WHITE SPACE
 000062                                 SR.LIP: .BLKW   2               ;ADDRESS OF LATEST APR
 000066                                 SR.LIN: .BLKW   2               ;ADDRESS OF LATEST IR
 000072                                 SR.GRS: .BLKW   3               ;GIVEN START DOC ID
 000100                                 SR.GRE: .BLKW   3               ;GIVEN END DOC ID
 000106                                 SR.ARS: .BLKW   3               ;CURRENT START DOC ID
 000114                                 SR.ARE: .BLKW   3               ;CURRENT END DOC ID
                                        ;
          000116                        DBSLEN=.-SR.YR                  ;LENGTH OF DB STATUS AREA
 000122                                 SR.LEN:                         ;LENGTH OF STATUS RECORD
                                        ;
                                        ;
                                        ; FILE DESCRIPTOR OFFSETS
                                        ;
 000000                                         .PSECT  FDSCOF,ABS
 000000                                 FD.FID: .BLKW   2               ;FILE ID
 000004                                 FD.FVR: .BLKW   1               ;VERSION NUMBER
 000006                                 FD.FNB: .BLKW   1               ;FILE NUMBER
 000010                                 FD.LEN:                         ;LENGTH OF FDSC
                                        ;
```

```
                                ;
                                ; M.MAC - MASTER COMPUTER PREFIX FILE
                                ;
                                ;
                                ;
                                ; MASTER COMPUTER CONSTANTS
                                ;
000043                          N.PKTS=35.              ;# OF FREE SSQ/QUQ PACKETS TO ALLOCATE
000020                          N.PKSZ=16.              ;SIZE OF PACKET (BYTES)
000240                          S.HRL=40.*N.BFAC        ;DEFAULT SIZE OF HRL FILE
                                ;
                                ;
                                ; MASTER SCHEDULER GLOBAL FLAGS
                                ;
000070                          M=56.
000000                          N=0
000004                          .REPT   N.BTCH
                                .IRP    Z,<\N>
                                CF.B'Z=M        ;DEFINE FLAG
                                .ENDR
                                M=M-1
                                N=N+2
                                .ENDR
                                ;
                                ; IN-CORE DHR GLOBAL FLAGS
                                ;
000000                          N=0
000002                          .REPT   N.SUNT
                                .IRP    Z,<\N>
                                CF.DR'Z=M       ;DEFINE FLAG
                                .ENDR
                                M=M-1
                                N=N+1
                                .ENDR
                                ;
                                ;
                                ; MASTER COMPUTER COMMAND SOURCES
                                ;
000000                          XMSCHED=0
000001                          XQT0=1
000002                          XHLMERG=2
000003                          XQTS=3
000004                          XDBLOAD=4
000005                          XSULOAD=5
000006                          XDMCIN=6
000007                          XFOSMRG=7
000010                          XHOTSK=8.
000011                          XHITSK=9.
000012                          XDBPROC=10.
000013                          XBATCH=11.
000014                          XGTSREC=12.
                                ;
                                ;
                                ; BATCH STATE STATUS FLAGS
                                ;
000000                          BS.INA=0        ;B.STTE: INACTIVE
000001                          BS.OPN=1        ;        OPEN
```

```
                000002          BS.GLS=2        ;       CLOSED
                000003          BS.SRC=3        ;       SEARCH
                000004          BS.DBU=4        ;       DATA BASE UPDATE
                                ;
                                ;
                                ; SEARCH UNIT STATUS FLAGS - SUST
                                ;
                000000          SU.IDL=0        ;IDLE
                000001          SU.LOD=1        ;CONTROL TABLE LOAD IN PROGRESS
                000002          SU.SRC=2        ;TABLES LOADED - SEARCHING
                000003          SU.XPD=3        ;SU XX DONE - WAIT FOR FOS
                000004          SU.DBU=4        ;FOS IN (SEARCH DONE) - DATA BASE UPDATE STARTED
                000005          SU.SRR=5        ;DBU DONE - STATUS RECORD REQUESTED
                000006          SU.DON=6        ;STATUS RECORD IN - SEARCH UNIT DONE
                                ;
                                ;
                                ; SEARCH UNIT STATUS ENTRY OFFSETS
                                ;
000000                                  .PSECT  SUSOFF,ABS
000000                          SS.STT: .BLKW   1       ;SEARCH UNIT STATUS
000002                          SS.FID: .BLKW   2       ;FID OF FOS.SPL
000006                          SS.FVR: .BLKW   1       ;VERSION
000010                          SS.FNB: .BLKW   1       ;FILE NAME
000012                          SS.LEN:                 ;SIZE OF ENTRY
                                ;
                                ;
                                ; DHR CONTROL WORD OFFSETS
                                ;
000000                                  .PSECT  DHROFF,ABS
000000                          DH.CTL: .BLKW   1       ;CONTROL WORD
000002                          DH.BF0: .BLKW   1       ;FIRST BUFFER OFFSET
000004                          DH.BF1: .BLKW   1       ;SECOND BUFFER OFFSET
000006                          DH.FLG: .BLKW   1       ;GLOBAL FLAG NUMBER
000010                          DH.DMC: .BLKW   1       ;DMCIN SAVE AREA
                                ;
                                ;
                                ; STATISTICS AREA OFFSETS - CHSTAT; CDSTAT; ETC
                                ;
000000                                  .PSECT  STTOFF,ABS
000000                          ST.BTC: .BLKW   1       ;# OF BATCHES
000002                          ST.QRY: .BLKW   1       ;QUERIES
000004                          ST.UHL: .BLKW   2       ;DOC/UHL
000010                          ST.HRL: .BLKW   2       ;HRL DOC'S
000014                          ST.XLT: .BLKW   2       ;XLATE TIME (TICKS)
000020                          ST.ASZ: .BLKW   2       ;FSA WORDS
000024                          ST.BSZ: .BLKW   2       ;FSB WORDS
000030                          ST.CSZ: .BLKW   2       ;FSC WORDS
000034                          ST.QSZ: .BLKW   2       ;QLS WORDS
000040                          ST.SCH: .BLKW   2       ;SEARCH TIME/SU (SEC)
000044                          ST.LEN:
                                ;
```

```
                                ;
                                ;
                                ; QUERY SPOOL FILE AREA OF BST OFFSETS
                                ;
                                        .PSECT  QSPLOF,ABS
                                Q.NQBK: .BLKW   1               ;NUMBER OF QUERY BLOCKS
                                Q.NUHL: .BLKW   1               ;NUMBER OF UHL BLOCKS
                                Q.FDSC: .BLKB   FD.LEN          ;FDSC OF QUERY.SPL
                                Q.SIZE:                         ;SIZE OF QUERY SPOOL FILE AREA
                                ;
                                ;
                                ;
                                ; BATCH STATUS TABLE OFFSETS TO BST0, BST1, ETC.
                                ;
                                        .PSECT  BSTOFF,ABS
                                        .BLKW   1               ;SSQ INTERLOCK
                                        .BLKW   1               ;SSQ ITEM COUNT
                                B.SSQ:  .BLKW   1               ;FORWARD LINK
                                        .BLKW   1               ;LAST ENTRY POINTER
                                B.SFDB: .BLKW   1               ;SCHEDULER'S FDB ADDRESS
                                B.SNDP: .BLKW   13.             ;SEND AREA
                                B.STAT: .BLKW   1               ;SCHEDULER STATUS FLAG
                                B.CNTX: .BLKW   1               ;STATE CONTEXT - NEXT/CURRENT NODE
                                B.SSQF: .BLKB   1               ;SSQ EVENT FLAG NUMBER
                                        .BLKB   1               ;NOT USED
                                B.NMBR: .BLKB   1               ;BATCH NUMBER
                                B.STTE: .BLKB   1               ;BATCH STATE
                                B.BSTA: .BLKW   1               ;BATCH STATE STATUS FLAGS
                                B.QUQP: .BLKW   1               ;UNLOADED QUQ ENTRY PRESENT FLAG
                                B.CQUQ: .BLKW   7               ;CURRENT QUQ ENTRY
                                B.QTTM: .BLKW   1               ;BATCH TRANSLATE TIME (TICKS)
                                B.FSAZ: .BLKW   1               ;FSA SIZE (WORDS)
                                B.FSBZ: .BLKW   1               ;FSB SIZE (WORDS)
                                B.FSCZ: .BLKW   1               ;FSC SIZE (WORDS)
                                B.QLSZ: .BLKW   1               ;QLS SIZE (WORDS)
                                B.UDOC: .BLKW   2               ;# OF DOC IN ALL UHL
                                B.HDOC: .BLKW   2               ;# OF DOC IN HRL
                                B.HBLK: .BLKW   1               ;# OF UNUSED BLOCKS IN HRL.MRG
                                B.HRLR: .BLKW   1               ;START BLOCK OF PREVIOUS HRL.MRG (READ)
                                B.HRLW: .BLKW   1               ;START BLOCK OF NEW HRL.MRG (WRITE)
                                B.HRLP: .BLKW   N.SUNT          ;HRL SUB-FILES PER SEARCH UNIT
                                B.FEMA: .BLKB   FD.LEN          ;FDSC FOR EMATRIX.EMA
                                B.FEMB: .BLKB   FD.LEN          ;FDSC FOR EMATRIX.EMB
                                B.FEMC: .BLKB   FD.LEN          ;FDSC FOR EMATRIX.EMC
                                B.FQLS: .BLKB   FD.LEN          ;FDSC FOR EMATRIX.QLS
                                B.FMHR: .BLKB   FD.LEN          ;FDSC FOR HRL.MRG
                                B.FFSA: .BLKB   FD.LEN          ;FDSC FOR TDCTA.FSA
                                B.FFSB: .BLKB   FD.LEN          ;FDSC FOR TDCTB.FSA
                                B.FFSC: .BLKB   FD.LEN          ;FDSC FOR TDCTC.FSA
                                B.NQRY: .BLKW   1               ;NUMBER OF QUERIES IN BATCH SO FAR
                                B.QMAP: .BLKW   N.QURY          ;MAP QID TO EQID
                                B.QSPL: .BLKB   Q.SIZE*N.QURY   ;QUERY SPOOL FILE AREA
                                B.SIZE:                         ;LENGTH OF BST
                                ;
```

```
                                ;
                                ;
                                ; QUERY ERROR CODES
                                ;
000144                          QE.RO1=100.     ;RESOURCE OVERFLOW TYPE 1
                                ;
```

```
                                ;
                                ; FILE NUMBERS
                                ;
000000                                  .PSECT  FNOFFS,ABS
000000                          FN.FSA: .BLKW   1       ;DP0:[7,5]TDCTA.FSA
000002                          FN.FSB: .BLKW   1       ;DP0:[7,5]TDCTB.FSA
000004                          FN.FSC: .BLKW   1       ;DP0:[7,5]TDCTC.FSA
000006                          FN.QLS: .BLKW   1       ;DP0:[7,5]EMATRIX.QLS
000010                          FN.MHR: .BLKW   1       ;DP0:[7,5]HRL.MRG
000012                          FN.EMA: .BLKW   1       ;DP0:[7,5]EMATRIX.EMA
000014                          FN.EMB: .BLKW   1       ;DP0:[7,5]EMATRIX.EMB
000016                          FN.EMC: .BLKW   1       ;DP0:[7,5]EMATRIX.EMC
000020                          FN.QRY: .BLKW   1       ;DP0:[7,4]QUERY.SPL
000022                          FN.DBS: .BLKW   1       ;DP0:[7,4]DBUPD.SPL
000024                          FN.MFO: .BLKW   1       ;DP0:[7,4]FOS.MRG
000026                          FN.DBR: .BLKW   1       ;DP0:[7,4]DBRSLT.SPL
000030                          FN.SF0: .BLKW   1       ;DP0:[300,1]FOS.SPL
000032                          FN.SF1: .BLKW   1       ;DP0:[300,2]FOS.SPL
000034                          FN.LGQ: .BLKW   1       ;DK1:[7,4]QUERY.SPL
000036                          FN.LGU: .BLKW   1       ;DK1:[7,4]DBUPD.SPL
000040                          FN.DHR: .BLKW   1       ;DP0:[7,4]DHR.SPL
000042                          FN.SHD: .BLKW   1       ;DP0:[7,4]MSCHED.SPL
000044                          FN.NMB:                 ;LENGTH OF TABLE
                                ;
                                ;
                                ;
                                ; STATE TRANSITION TABLE OFFSETS
                                ;
000000                                  .PSECT  WNODOF,ABS      ;WAIT NODE
000000                          WN.SRC: .BLKB   1       ;COMMAND SOURCE
000001                          WN.TYP: .BLKB   1       ;COMMAND TYPE
000002                          WN.ROT: .BLKW   1       ;HANDLING ROUTINE
000004                          WN.NTP: .BLKW   1       ;NEXT NODE TYPE
000006                          WN.NXT: .BLKW   1       ;NEXT STATE
000010                          WN.SIZ:                 ;SIZE OF NODE ENTRY
                                ;
                                ;
000000                                  .PSECT  DNODOF,ABS      ;DECISION NODE
000000                          DN.DCK: .BLKW   1       ;DECISION CHECK ROUTINE
000002                          DN.ROT: .BLKW   1       ;DECISION SATISFIED ROUTINE
000004                          DN.NTP: .BLKW   1       ;NEXT NODE TYPE
000006                          DN.NXT: .BLKW   1       ;NEXT STATE
000010                          DN.SIZ:                 ;SIZE OF NODE ENTRY
                                ;
                                ;
```

```
    219 000000                          .PSECT
```

```
      1         000001                          .END
```

SYMBOL TABLE

```
BITVAL= 000000          B.FFSC  000222      010 DN.ROT  000002      013 Q.SIZE  000014      007 ST.UHL  000004      006
BIT0  = 000001          B.FMHR  000172      010 DN.SIZ  000010      013 SR.ARE  000114      002 ST.XLT  000014      006
BIT1  = 000002          B.FQLS  000162      010 FD.FID  000000      003 SR.ARS  000106      002 SU.DBU= 000004
BIT10 = 002000          B.FSAZ  000100      010 FD.FNB  000006      003 SR.DAY  000010      002 SU.DON= 000006
BIT11 = 004000          B.FSBZ  000102      010 FD.FVR  000004      003 SR.DLT  000014      002 SU.IDL= 000000
BIT12 = 010000          B.FSCZ  000104      010 FD.LEN  000010      003 SR.ECB  000047      002 SU.LOD= 000001
BIT13 = 020000          B.HBLK  000120      010 FN.DBR  000026      011 SR.ECH  000046      002 SU.SRC= 000002
BIT14 = 040000          B.HDOC  000114      010 FN.DBS  000022      011 SR.ECL  000050      002 SU.SRR= 000005
BIT15 = 100000          B.HRLP  000126      010 FN.DHR  000040      011 SR.FIB  000012      002 SU.XPD= 000003
BIT2  = 000004          B.HRLR  000122      010 FN.EMA  000012      011 SR.GRE  000100      002 S.HRL = 000240
BIT3  = 000010          B.HRLW  000124      010 FN.EMB  000014      011 SR.GRS  000072      002 WN.NTP  000004      012
BIT4  = 000020          B.NMBR  000052      010 FN.EMC  000016      011 SR.LEN  000122      002 WN.NXT  000006      012
BIT5  = 000040          B.NQRY  000232      010 FN.FSA  000000      011 SR.LIN  000066      002 WN.ROT  000002      012
BIT6  = 000100          B.QLSZ  000106      010 FN.FSB  000002      011 SR.LIP  000062      002 WN.SIZ  000010      012
BIT7  = 000200          B.QMAP  000234      010 FN.FSC  000004      011 SR.MON  000006      002 WN.SRC  000000      012
BIT8  = 000400          B.QSPL  000316      010 FN.LGQ  000034      011 SR.NDC  000042      002 WN.TYP  000001      012
BIT9  = 001000          B.QTTM  000076      010 FN.LGU  000036      011 SR.NDS  000036      002 WORD0 = 000000
BS.CLS= 000002          B.QUQP  000056      010 FN.MFD  000024      011 SR.NIN  000030      002 WORD1 = 000002
BS.DBU= 000004          B.SFDB  000010      010 FN.MHR  000010      011 SR.NIP  000022      002 WORD2 = 000004
BS.INA= 000000          B.SIZE  000772      010 FN.NMB  000044      011 SR.SDB  000032      002 WORD3 = 000006
BS.OPN= 000001          B.SNDP  000012      010 FN.OLS  000006      011 SR.SRC  000002      002 WORD4 = 000010
BS.SRC= 000003          B.SSQ   000004      010 FN.QRY  000020      011 SR.SUN  000000      002 WORD5 = 000012
BYTE0 = 000000          B.SSQF  000050      010 FN.SF0  000030      011 SR.TWS  000056      002 WORD6 = 000014
BYTE1 = 000001          B.STAT  000044      010 FN.SF1  000032      011 SR.WSL  000052      002 WORD7 = 000016
BYTE2 = 000002          B.STTE  000053      010 FN.SHD  000042      011 SR.YR   000004      002 WORD8 = 000020
BYTE3 = 000003          B.UDOC  000110      010 M       = 000062        SR.1IN  000024      002 WORD9 = 000022
BYTE4 = 000004          CF.B0 = 000070          N       = 000002        SR.1IP  000016      002 WRDVAL= 000024
BYTE5 = 000005          CF.B2 = 000067          N.BFAC= 000004          SS.FID  000002      004 XBATCH= 000013
BYTE6 = 000006          CF.B4 = 000066          N.BHGH= 000006          SS.FNB  000010      004 XDALOA= 000004
BYTE7 = 000007          CF.B6 = 000065          N.BTCH= 000004          SS.FVR  000006      004 XDBPRO= 000012
BYTE8 = 000010          CF.DR0= 000064          N.BUFB= 004000          SS.LEN  000012      004 XDMCIN= 000006
BYTE9 = 000011          CF.DR1= 000063          N.BUFW= 002000          SS.STT  000000      004 XFOSMR= 000007
BYTVAL= 000012          DBSLEN= 000116          N.FOS = 000764          ST.ASZ  000020      006 XGTSRE= 000014
B.BSTA  000054      010 DH.BF0  000002      005 N.PKSZ= 000020          ST.BSZ  000024      006 XHITSK= 000011
B.CNTX  000046      010 DH.BF1  000004      005 N.PKTS= 000043          ST.BTC  000000      006 XHLMER= 000002
B.CQUQ  000060      010 DH.CTL  000000      005 N.QURY= 000031          ST.CSZ  000030      006 XHOTSK= 000010
B.FEMA  000132      010 DH.DMC  000010      005 N.SUNT= 000002          ST.HRL  000010      006 XMSCHE= 000000
B.FEMB  000142      010 DH.FLG  000006      005 QE.RQ1= 000144          ST.LEN  000044      006 XQTS  = 000003
B.FEMC  000152      010 DN.DCK  000000      013 Q.FDSC  000004      007 ST.QRY  000002      006 XQT0  = 000001
B.FFSA  000202      010 DN.NTP  000004      013 Q.NQBK  000000      007 ST.QSZ  000034      006 XSULOA= 000005
B.FFSB  000212      010 DN.NXT  000006      013 Q.NUHL  000002      007 ST.SCH  000040      006

. ABS.  000000      000
        000000      001
SRCOFF  000122      002
FDSCOF  000010      003
SUSOFF  000012      004
DHROFF  000012      005
STTOFF  000044      006
QSPLOF  000014      007
BSTOFF  000772      010
FNOFFS  000044      011
WNODOF  000010      012
DNODOF  000010      013
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  2015 WORDS  ( 8 PAGES)
DYNAMIC MEMORY:  3068 WORDS  ( 11 PAGES)
```

```
.MAIN.  MA    M1110  27-MAR-80 13:54  PAGE 11-2
SYMBOL TABLE

ELAPSED TIME:  00:00:14
.M/-SP=LIST,P,M,END
```

TABLE OF CONTENTS

10- 3 INTRODUCTION
11- 70 MACROS
12- 98 EQUATED SYMBOLS
13- 109 DATA BUFFERS

```
                        .TITLE  HOST INPUT TASK (HITSK)
                        .IDENT  /01/
                        .SBTTL  INTRODUCTION
                ;--
                ;
                ; RECEIVE INPUT FROM THE HOST PROCESSOR
                ;
                ; THIS TASK IS RESPONSIBLE FOR INPUTTING FILES OF DATA FROM THE
                ; HOST PROCESSOR AND AND SPOOLING THEM TO THE SYSTEM DISK ON THE
                ; MASTER COMPUTER.  THERE ARE TWO BASIC TYPES OF FILES THAT ARE
                ; SPOOLED:
                ;
                ;       1)      QUERIES AND USER HIT LISTS
                ;       2)      DATA BASE UPDATE FILES
                ;
                ; THE QUERIES AND HITS LISTS ARE SPOOLED TO [7,5]QUERY.SPL AND THE
                ; DATA BASE UPDATES ARE SPOOLED TO [7,5]DBUPD.SPL.
                ;
                ; A GIVEN QUERY/HIT LIST TRANSACTION CONTAINS FIRST THE QUERIES AND
                ; THEN THE HIT LISTS.  THESE ARE BOTH PLACED INTO THE SAME FILE BY
                ; THE "QRY" ROUTINE.
                ;
                ; DATA BASE UPDATES CONSISTS OF SEVEN TYPES OF RECORDS:  THESE ARE:
                ;
                ;       1)      NEW DOCUMENT
                ;       2)      MODIFY DOCUMENT
                ;       3)      OVERLAY DOCUMENT
                ;       4)      ADD DOCUMENT
                ;       5)      PURGE DOCUMENT
                ;       6)      READ DOCUMENT
                ;       7)      DELETE DOCUMENT
                ;
                ; THE NEW, MODIFY, OVERLAY, AND ADD DOCUMENT RECORDS ARE HANDLED BY
                ; THE SAME ROUTINE "DOC".  THE PURGE, READ AND DELETE DOCUMENT RECORDS
                ; ARE HANDLED BY THE "PURGE" ROUTINE.  THE DATA BASE UPDATE RECORDS HAVE
                ; BEEN GROUPED INTO BEING PROCESSED BY THESE TWO ROUTINES BECAUSE OF
                ; THE SIMILARITY OF PROCESSING REQUIREMENTS.
                ;
                ; THE GENERAL PROCESSING FLOW IS AS FOLLOWS:
                ;
                ;       1)      THE CHANNEL IS INITIALIZED.  IF AT ANY TIME AN
                ;               ERROR IS DETECTED, THE PROCESSING RETURNS TO
                ;               THIS POINT.
                ;
                ;       2)      A FUNCTION IS RECEIVED.
                ;
                ;       3)      THE FUNCTION IS PARSED BY THE "PARSE" ROUTINE
                ;
                ;       4)      "DSPTCH" IS CALLED TO DETERMINE WHICH ROUTINE
                ;               IS TO BE CALLED TO PROCESS THE INCOMMING FILE.
                ;
                ;       5)      THE ROUTINE IS CALLED.  WHILE IN THE ROUTINE
                ;               THE INPUT FILE IS SPOOLED.
                ;
                ;       6)      AFTER THE ROUTINE IS FINISHED CONTROL IS RETURNED
                ;               WAIT FOR ANOTHER FUNCTION.
                ;
```

```
                        ;
                        ; DATE WRITTEN: FEB 24,1979
                        ;
                        ; DATE MODIFIED:
                        ;
                        ; GENERAL REGISTER USAGE:
                        ;
                        ;       R5 = BLOCK COUNT
                        ;       R4 = OPTIONAL DATA
                        ;       R3 = BYTE LENGTH OF EXCHANGE
                        ;       R2 = INPUT PACKING MODE
                        ;
```

```
                        .SBTTL  MACROS
                ;
                ; MACROS
                ;
                        .MCALL  QIOW$S,WRITE$,WAIT$,OFNB$W,FINIT$
                        .MCALL  FDBDF$,FDRC$A,FDBK$A,FSRSZ$,FDOP$A
                        .MCALL  CLOSE$,NMBLK$
                .MACRO  MSGSTR,STRING,?L1,?L2
                        .WORD   L2-L1
                        .WORD   L1
                L1:
                        .ASCII  /STRING/
                L2:
                        .EVEN
                .ENDM
                ;
                ;
                .MACRO  ENT,TYPE,ROUTIN,PACK,FID,TXLGT,?L1,?L2
                L1:
                        .WORD   TYPE            ;EXCHANGE TYPE
                        .WORD   ROUTIN          ;ROUTINE TO BE CALLED
                        .WORD   PACK            ;PACKING MODE
                        .WORD   FID             ;FILE ID
                        .WORD   TXLGT           ;BYTE LENGTH OF EXCHANGE
                L2:
                TBLSZ$  =       L2-L1
                .ENDM
```

```
                                .SBTTL  EQUATED SYMBOLS
                        ;
                        ; EQUATED SYMBOLS
                        ;
000001                  LUN1    =       1
000001                  EF.1    =       1
000002                  WRT.EF  =       2
000002                  WRT.LU  =       2               ;SPOOL FILE LUN
000003                  WRT.DK  =       3               ;EXCHANGE LOG FILE LUN
000006                  FUCLEN  =       6               ;INPUT FUNCTION LENGTH IN BYTES
```

```
                                        .SBTTL  DATA BUFFERS

                                ;
                                ; DATA BUFFERS
                                ;
                                ; INPUT DATA BUFFERS
                                ;
000000                          INBUF:  .BLKB   N.BUFB          ;INPUT DATA BUFFER
                                ;
                                ; MAIN ROUTINE DISPATCH TABLE
                                ;
004000                          TBL:    ENT     5,QHL,3,FN.ORY,2048.    ;QUERY/UHL
004012                                  ENT     6,DOC,3,FN.DBS,2048.    ;NEW DOCUMENT
004024                                  ENT     7.PURGE,4.FN.DBS,2046.  ;PURGE DOC.
004036                                  ENT     8.,DOC,3.FN.DBS,2048.   ;MODIFY-REPLACE
004050                                  ENT     9.,DOC,3.FN.DBS,2048.   ;MODIFY-OVERLAY
004062                                  ENT     11.,PURGE,4.FN.DBS,2046.;READ DOC.
004074                                  ENT     13.,DOC,3.FN.DBS,2048.  ;ADD SUB-DOC.
004106                                  ENT     14.,PURGE,4.FN.DBS,2046.;DELETE SUB-DOC.
004120                                  ENT     16.,QHL,2,FN.ORY,32.    ;REQUEST MASS UPDATE
004132                                  ENT     18.,QHL,2,FN.DBS,32.    ;END MASS UPDATE
004144                                  ENT     0.0.0.0.0
                                ;
                                ; I/O STATUS BLOCKS
                                ;
004156                          IOSTAT: .BLKW   2       ;GET FUNCTION I/O STATUS BLOCK
004162                          WRSTST: .BLKW   2       ;WRITE BLOCK I/O STATUS
004166                          QIOST:  .BLKW   2       ;READ INPUT DATA BUFFER I/O STATUS BLOCK
                                ;
                                ;   FLAG TO INDICATE WHETHER DK1: IS ONLINE (LOGFLG = 0), OR
                                ;   OFFLINE (LOGFLG = 1)
                                ;
004172  000000                  LOGFLG: .WORD   0
                                ;
                                ; SPOOL FILE CONTROL BLOCK
                                ;
004174                          FDB:    FDBDF$
004334                                  FDRC$A  FD.RWM
004334                                  FDBK$A  INBUF,N.BUFB,,WRT,EF,WRSTST
004334                                  FDOP$A  WRT,LU
                                ;
                                ;   EXCHANGE LOG FILE CONTROL BLOCK
                                ;
004334                          FDBLOG: FDBDF$
004474                                  FDRC$A  FD.RWM
004474                                  FDBK$A  INBUF,N.BUFB,,WRT,EF,WRSTST
004474                                  FDOP$A  WRT,DK
                                ;
004474                                  FSRSZ$  0
                                ;
                                ;   DUMMY NAMEBLOCK TO GET DIRECTORY FID FOR LOG FILE
                                ;
004474                          LOGNBK: NMBLK$  ,,,DK,1
                                ;
                                ;   FILE NAME/TYPE INDEX
                                ;
004532  004536'                 FNINDX: .WORD   LGQNAM           ;FN.LGQ
004534  004546'                         .WORD   LGUNAM           ;FN.LGU
```

```
                                        ;
004536  066615  072150  000000  LGQNAM: .RAD50  /QUERY   SPL/  ;FN.LGQ
004544  074514
004546  014545  062240  000000  LGUNAM: .RAD50  /DBUPD   SPL/  ;FN.LGU
004554  074514
                                        ;
                                        ;   DEVICE NAME INDEX
                                        ;
004556      104     113         DVINDX: .ASCII  /DK/           ;FN.LGQ
004560      104     113                 .ASCII  /DK/           ;FN.LGU
                                        ;
                                        ;   DEVICE UNIT INDEX
                                        ;
004562  000001                    UNINDX: .WORD   1              ;FN.LGQ
004564  000001                            .WORD   1              ;FN.LGU
                                        ;
                                        ;   DIRECTORY NAME DESCRIPTION
                                        ;
004566  000005                    DIRDSi: .WORD   5
004570  004572'                           .WORD   DIRDTi
004572      133     067     054 DIRDTi: .ASCII  /[7,4]/
004575      064     135
                                        .EVEN
                                        ;
                                        ;   DIRECTORY FID
                                        ;
004600                          DIRK74: .BLKW   3              ;DKi:[7,4]
                                        ;
                                        ; ERROR MESSAGES
                                        ;
004606                          RCVERR: MSGSTR  <ACC RECEIVE ERROR = %D, PC = %O>
004652                          CPERR:  MSGSTR  <FILE OPEN ERROR = %D, PC = %O>
004714                          WRTERR: MSGSTR  <FILE WRITE ERROR = %D, PC = %O>
004756                          DIR:    MSGSTR  <DIRECTIVE ERROR = %D, PC = %O>
005020                          IRT:    MSGSTR  <INVALID RECORD TYPE>
005050                          CFFLIN: MSGSTR  <EXCHANGE LOGGING DEVICE (DKi:) NOT MOUNTED>
005126                          LOGERR: MSGSTR  <LOG DEVICE ERROR %I, PC = %O -- LOG DISABLED>
                                        ;
                                        ; ERROR PARAMETER BUFFER
                                        ;
005206                          PAR:    .BLKW   10.
```

```
                                        ;+
                                        ;
                                        ; **--MAIN ROUTINE
                                        ;
                                        ; THIS IS THE MAIN ROUTINE OF THE "HITSK" TASK.  THIS ROUTINE
                                        ; INITIALIZES THE CHANNEL TO THE UNIVAC HOST COMPUTER,
                                        ; GETS A FUNCTION WORD, PARSES IT, AND CALLS THE APPROPRIATE
                                        ; PROCESSING ROUITNE.
                                        ;
                                        ; INPUT:
                                        ;       NONE
                                        ;
                                        ; OUTPUT:
                                        ;       NONE
                                        ;
                                        ;-
005232                                  MAIN:
005232                                          FINIT$                          ;INITIALIZE THE FILE SYSTEM
                                        ;
                                        ;   INITIALIZE THE FID OF THE LOG FILES' DIRECTORY
                                        ;
005236  005067  176730                          CLR     LOGFLG
005242  012700  004334'                         MOV     #FDBLOG,R0
005246  012701  004474'                         MOV     #LOGNBK,R1
005252  012702  004566'                         MOV     #DIRDS1,R2
005256                                          CALL    .GTDIR                  ;GET DIRECTORY FID
005262  012702  004600'                         MOV     #DIRK74,R2
005266  016712  177226                          MOV     LOGNBK+N.DID,(R2)
005272  016762  177224  000002                  MOV     LOGNBK+N.DID+2,2(R2)
005300  016762  177220  000004                  MOV     LOGNBK+N.DID+4,4(R2)
005306  005762  000004                          TST     4(R2)                   ;IS DK1 ONLINE?
005312  001411                                  BEQ     INL                     ;BRANCH IF YES
005314  005267  176652                          INC     LOGFLG                  ;SET OFFLINE FLAG
005320                                          MOUT$S  #OFFLIN
005336                                  INL:
005336                                          QIOW$S  #IO.INL,#LUN1,#EF,1,,#IOSTAT ;INITIALIZE THE CHANNEL
005404  105767  176546                          TSTB    IOSTAT                  ;IS THERE AN ERROR?
005410  002752                                  BLT     INL                     ;YES, GO RE-INITIALIZE
005412                                  MORE:
005412                                          QIOW$S  #IO.RTC,#LUN1,#EF,1,,#IOSTAT,,<#INBUF,#FUCLEN,#6>
005466  105767  176464                          TSTB    IOSTAT
005472  002721                                  BLT     INL                     ;THERE IS AN ERROR, REINITIALIZE
005474                                          CALL    PARSE                   ;PARSE THE INPUT FUNCTION
005500                                          CALL    DSPTCH                  ;DETERMINE WHICH ROUTINE IS TO PROCESS
005504                                          CALL    (R0)                    ;CALL THE ROUTINE
005506                                          CLOSE$  #FDB
005516                                          CLOSE$  #FDBLOG
005526  000731                                  BR      MORE                    ;GO GET ANOTHER FUNCTION
                                        ;+
                                        ;
                                        ; **--PARSE ROUTINE
                                        ;
                                        ; THIS ROUTINE TAKES AN INPUT FUNCTION AND PARSES IT INTO ITS CONSTITUANT
                                        ; PARTS.  IN EVERY FUNCTION THERE ARE THREE  PARTS:
                                        ;
                                        ;       1)      OPTIONAL DATA IN BITS 0-17
                                        ;       2)      BLOCK COUNT IN BITS 18-29, AND
```

```
                                ;       3)      TRANSACTION TYPE IN BITS 30-35.
                                ;
                                ; INPUT:
                                ;       NONE
                                ;
                                ; OUTPUT:
                                ;       R5 = BLOCK COUNT
                                ;       R4 = OPTIONAL DATA
                                ;       R3 = TRANSACTION TYPE
                                ;
                                ;-
                                ;
005530                          PARSE:
005530  016704  172250                  MOV     INBUF+4,R4              ; OPTIONAL DATA
005534  016705  172242                  MOV     INBUF+2,R5              ;GET THE BLOCK COUNT
005540  042705  140000                  BIC     #140000,R5
005544  072527  177776                  ASH     #-2,R5                  ;R5 = BLOCK COUNT
005550  016700  172224                  MOV     INBUF,R0                ;GET THE TRANSACTION TYPE
005554  016701  172222                  MOV     INBUF+2,R1
005560  073027  000002                  ASHC    #2,R0
005564  010003                          MOV     R0,R3                   ;R3 = TRANSACTION TYPE
005566  000207                          RETURN
                                ;+
                                ;
                                ; **--DSPTCH ROUTINE
                                ;
                                ; THIS ROUTINE DETERMINES WHICH ROUTINE IS TO PROESS A GIVEN
                                ; TRANACTION TYPE.
                                ;
                                ; INPUT:
                                ;       R5 = BLOCK COUNT
                                ;       R4 = OPTIONAL DATA
                                ;       R3 = TRANSACTION TYPE
                                ;
                                ; OUTPUT:
                                ;       R0 = ADDRESS OF ROUTINE TO BE CALLED
                                ;       R1 = FID
                                ;       R2 = ACC PACKING MODE
                                ;       R3 = BYTE LENGTH OF BLOCK COMING FROM HOST
                                ;-
                                ;
005570                          DSPTCH:
005570  012700  004000'                 MOV     #TBL,R0         ;GET THE ADDRESS OF THE DISPATCH TABLE
005574                          1$:
005574  005710                          TST     (R0)            ;IS THIS THE END OF THE TABLE.?
005576  001416                          BEQ     50$             ;YES, GO SET UP FOR ERROR CALL
005600  020310                          CMP     R3,(R0)         ;NO, IS THIS THE CORRECT ENTRY.?
005602  001011                          BNE     2$              ;NO, GO GET NEXT ENTRY
005604  016002  000004                  MOV     4(R0),R2        ;YES, GET THE PACKING MODE
005610  016001  000006                  MOV     6(R0),R1        ;GET THE FILE ID (FID)
005614  016003  000010                  MOV     10(R0),R3       ;R3 = BYTE LENGTH OF EXCHANGE
005620  016000  000002                  MOV     2(R0),R0        ;GET THE ADDRESS OF THE ROUTINE TO BE CALLED
005624  000405                          BR      99$             ;GO EXIT
005626                          2$:
005626  062700  000012                  ADD     #TBLSZ$,R0      ;POINT R0 AT THE NEXT ENTRY
005632  000760                          BR      1$              ;GO LOOK AT THE NEXT ENTRY
005634                          50$:
```

```
                                        MOV     #ERROR,R0       ;GET READY TO CALL THE ERROR ROUTINE
                                99$:
                                        RETURN
                                ;+
                                ;
                                ; THIS ROUTINE PROCESSES THE ADD, MODIFY AND OVERLAY DOCUMENT
                                ; AND THE ADD SUB-DOCUMENT TRANSACTION TYPES.
                                ;
                                ; INPUT:
                                ;       GENERAL REGISTER SET
                                ;
                                ; OUTPUT:
                                ;       R5 = 0 IF EVERYTHING WENT OK
                                ;-
                                ;
                                ;
                                IOC:
                                        MOV     #FDB,R0         ;R0->FDB
                                        CALL    BLDNFL          ;BUILD THE FILE NAME BLOCK
                                        CALL    OPEN            ;OPEN THE FILE
                                        BCS     50$             ;IF ERROR GO DELETE THE FILE
                                1$:
                                        CALL    RECEVE          ;GET A BLOCK OF THE INPUT TRANSACTION
                                        BCS     50$             ;IF ERROR DELETE FILE AND RETURN
                                        CALL    WRITE           ;WRITE THE SPOOL FILE
                                        BCS     50$             ;IF ERROR DELETE FILE AND RETURN
                                        SOB     R5,1$           ;CONTINUE UNTIL ALL BLOCKS INPUT
                                        BR      99$             ;GO EXIT
                                50$:
                                        MOV     #FDB,R0         ;R0 -> FDB OF SPOOL FILE
                                        CALL    .DLFNB          ;ERROR DELETE FILE
                                99$:
                                        RETURN
                                ;+
                                ;
                                ; THIS ROUTINE HANDLES THE DELETE, READ, AND PURGE DOCUMENT AND
                                ; THE DELETE SUB-DOCUMENT TRANSACTIONS FROM THE HOST.
                                ;
                                ; INPUT:
                                ;       GENERAL REGISTER USAGE
                                ;
                                ; OUTPUT:
                                ;       R5=0 IF ALL BLOCK SUCCESSFULLY INPUT
                                ;
                                ;-
                                ;
                                PURGE:
                                        MOV     #FDB,R0                 ;R0->FILE CONTROL BLOCK
                                        CALL    BLDNFL                  ;BUILD A FILE NAME BLOCK
                                        CALL    OPEN                    ;OPEN THE FILE
                                        BCS     99$                     ;IF ERROR EXIT
                                        CALL    RECEVE                  ;GET THE FIRST BLOCK
                                        BCS     50$                     ;IF ERROR CLOSE FILE AND EXIT
                                        MOV     INBUF+2,INBUF           ;PUT IN HEADER CHARACTERS
                                        CLR     INBUF+2
                                        CALL    WRITE                   ;WRITE THE BLOCK
                                        BCS     50$                     ;IF ERROR CLOSE FILE AND EXIT
```

```
376 005756  005305                        DEC     R5                      ;DEC INPUT BLOCK COUNT
377 005760  001407                        BEQ     2$                      ;IF NO MORE BLOCKS CLOSE AND EXIT
378 005762.                         1$:
379 005762.                               CALL    RECEVE                  ;GET THE NEXT BLOCK
380 005766  103407                        BCS     50$                     ;IF ERROR DELETE FILE AND EXIT
381 005770                                CALL    WRITE                   ;WRITE THE NEXT BLOCK
382 005774  103404                        BCS     50$                     ;IF ERROR, DELETE FILE AND EXIT
383 005776  077507                        SOB     R5,1$                   ;CONTINUE UNTIL ALL BLOCKS INPUT
384 006000                          2$:
385 006000                                CLOSE$                          ;ERROR, CLOSE FILE AND EXIT
386 006004  000402.                       BR      99$
387 006006                          50$:
388 006006                                CALL    .DLFNB                  ;ERROR, DELETE FILE AND EXIT
389 006012.                         99$:
390 006012  000207                        RETURN
391                                 ;+
392.                                ;
393                                 ; **--QHL ROUTINE
394                                 ;
395                                 ; THIS ROUTINE PROCESSES THE QUERY/HIT LIST TRANSACTIONS
396                                 ; THE QUERIES ARE INPUT IN ONE FORMAT AND THE HIT LISTS ARE INPUT
397                                 ; IN ANOTHER FORMAT.
398                                 ;
399                                 ; INPUTS:
400                                 ;       GENERAL REGISTER USAGE
401                                 ;
402.                                ; OUTPUTS:
403                                 ;       NONE
404                                 ;
405 006014                          QHL:
406 006014  012700  004174'               MOV     #FDB,R0                 ;R0->FILE CONTROL BLOCK
407 006020                                CALL    BLDNFL                  ;BUILD A FILE NAME BLOCK
408 006024                                CALL    OPEN                    ;OPEN THE FILE
409 006030  103456                        BCS     99$                     ;IF ERROR EXIT
410 006032. 160405                        SUB     R4,R5                   ;R4=NO OF QUERY BLOCKS, R5=NO OF HRL BLOCKS
411 006034                                CALL    RECEVE                  ;GET THE FIRST BLOCK
412.006040  103503                        BCS     50$                     ;IF ERROR DELETE THE FILE AND EXIT
413 006042. 010467  171734                MOV     R4,INBUF+2              ;PUT QUERY BLOCK COUNT INTO BUFFER
414 006046                                CALL    WRITE                   ;WRITE THE BLOCK TO THE SPOOL FILE
415 006052. 103476                        BCS     50$                     ;IF ERROR, DELETE THE FILE AND EXIT
416 006054                                CALL    CONVRT                  ;CONVERT ASCII QID TO BINARY
417 006060  010446                        MOV     R4,-(SP)                ;SAVE R4 AND R5 FOR BUILDING THE...
418 006062. 010546                        MOV     R5,-(SP)                ;QUQ CONTROL BLOCK
419 006064  005304                        DEC     R4                      ;DEC INPUT QUERY BLOCK COUNT
420 006066  001407                        BEQ     2$                      ;IF NO MORE QUERY BLOCKS, GET HRL'S
421 006070                          1$:
422.006070                                CALL    RECEVE                  ;GET NEXT QUERY BLOCK
423 006074  103465                        BCS     50$                     ;IF ERROR, DELETE FILE AND EXIT
424 006076                                CALL    WRITE                   ;WRITE THE FILE
425 006102. 103462.                       BCS     50$                     ;IF ERROR, DELETE FILE AND EXIT
426 006104  077407                        SOB     R4,1$                   ;GO GET ANOTHER QUERY BLOCK
427 006106                          2$:
428.                                ;
429                                 ; INPUT HRL BLOCKS
430                                 ;
431 006106  005705                        TST     R5                      ;ANY UHL'S?
432.006110  003426                        BLE     99$                     ;NO, EXIT
```

```
 433 006112  012702  000004                    MOV     #4,R2           ;CHANGE TO PACKING MODE 4
 434 006116  012703  003776                    MOV     #2046.,R3       ;R3 = INPUT BLOCK'S BYTE LENGTH
 435 006122                                    CALL    RECEVE          ;GET FIRST BLOCK OF HRL'S
 436 006126  103450                            BCS     50$             ;IF ERROR, DELETE FILE AND EXIT
 437 006130  016767  171646  171642            MOV     INBUF+2,INBUF   ;PUT IN PROPER BLOCK ID
 438 006136                                    CALL    WRITE           ;WRITE TO THE SPOOL FILE
 439 006142  103442                            BCS     50$             ;IF ERROR, DELETE FILE AND EXIT
 440 006144  005305                            DEC     R5              ;DEC INPUT HRL BLOCK COUNT
 441 006146  001407                            BEQ     99$             ;IF ALL INPUT CLOSE FILE AND EXIT
 442 006150                              3$:
 443 006150                                    CALL    RECEVE          ;GET NEXT HRL BLOCK
 444 006154  103435                            BCS     50$             ;IF ERROR, DELETE FILE AND EXIT
 445 006156                                    CALL    WRITE           ;WRITE BLOCK TO SPOOL FILE
 446 006162  103432                            BCS     50$             ;IF ERROR, DELETE FILE AND EXIT
 447 006164  077507                            SOB     R5,3$           ;CONTINUE UNTIL ALL BLOCKS INPUT
 448 006166                              99$:
 449 006166                                    CLOSE$                  ;CLOSE FILE
 450                                     ;
 451                                     ; BUILD THE CONTROL BLOCK THAT IS SENT TO MSCHED
 452                                     ;
 453 006172                                    CALL    GETFRE          ;GET A PACKET FROM COMMON
 454 006176  012662  000004                    MOV     (SP)+,4.(R2)    ;GET THE NO. OF HRL BLOCKS
 455 006202  012662  000002                    MOV     (SP)+,2.(R2)    ;GET THE NO. OF ORY BLOCKS
 456 006206  010162  000006                    MOV     R1,6.(R2)       ;GET THE EQID
 457 006212  016062  000102  000010            MOV     F.FNB+N.FID(R0),8.(R2)  ;GET THE FILE ID
 458 006220  016062  000104  000012            MOV     F.FNB+N.FID+2(R0),10.(R2);
 459 006226  016062  000120  000014            MOV     F.FNB+N.FVER(R0),12.(R2);GET THE FILE VERSION NO.
 460 006234  012762  000020  000016            MOV     #FN.ORY,14.(R2) ;GET THE FILE NUMBER
 461 006242                                    CALL    PUTQUB          ;SEND CONTROL BLOCK TO MSCHED
 462 006246  000207                            RETURN                  ;EXIT
 463 006250                              50$:
 464 006250                                    CALL    .DLFNB          ;DELETE FILE
 465                                     ;      MOV     (SP)+,(SP)+     ;CLEAN UP THE STACK
 466 006254  000207                            RETURN                  ;EXIT
 467                                     ;+
 468                                     ;
 469                                     ; **--RECEVE ROUTINE
 470                                     ;
 471                                     ; THIS ROUTINE INPUTS A BLOCK FROM THE ACC LINK
 472                                     ;
 473                                     ; INPUTS:
 474                                     ;       GENERAL REGISTER USAGE
 475                                     ;
 476                                     ; OUTPUTS:
 477                                     ;       IF AN ERROR WAS DETECTED THE "C" BIT WILL BE SET, ELSE IT
 478                                     ;       IS CLEARED.
 479                                     ;
 480                                     ;-
 481                                     ;
 482 006256                              RECEVE:
 483 006256  010446                            MOV     R4,-(SP)                ;SAVE R4
 484 006260                                    QIOW$S  #IO.RLB,#LUN1,#EF,1,,#QIOST,,<#INBUF,R3,R2>
 485 006330  105767  175632                    TSTB    QIOST
 486 006334  003021                            BGT     90$                     ;NO ERROR, RETURN
 487 006336  116704  175624                    MOVB    QIOST,R4                ;GET ERROR CODE
 488 006342  010467  176640                    MOV     R4,PAR                  ;INTO ERROR PARAMETER
 489 006346  016667  000002  176634            MOV     2(SP),PAR+2             ;GET CALLING LOCATION
```

```
                                        MOUT$S  #RCVERR,#PAR            ;ISSUE ERROR MSG
                                        SEC
                                        BR      99$
                                90$:
                                        CLC
                                99$:
                                        MOV     (SP)+,R4                ;RESTORE R4
                                        RETURN
                                ;+
                                ;
                                ; **--OPEN ROUTINE
                                ;
                                ; THIS ROUTINE OPENS THE CORRECT SPOOL FILE AND THE CORRECT EXCHANGE LOG FILE
                                ;
                                ; INPUTS:
                                ;       GENERAL REGISTER USAGE, AND R0->FDB
                                ;
                                ; OUTPUTS:
                                ;       IF ERROR THE "C" BIT IS SET, ELSE IT IS CLEARED
                                ;-
                                ;
006406  010046                  OPEN:   MOV     R0,-(SP)                ;SAVE R0
006410  010446                          MOV     R4,-(SP)                ;SAVE R4
006412                                  OFNB$W                          ;OPEN FILE FOR WRITTING
006424  105760  000052                  TSTB    F.ERR(R0)               ;IS THERE AN ERROR?
006430  003451                          BLE     90$                     ;NO, RETURN
006432  005767  175534                  TST     LOGFLG                  ;IS DK1 OFFLINE?
006436  001067                          BNE     93$                     ;BRANCH IF YES
                                ;
                                ;   OPEN THE CORRECT EXCHANGE LOG FILE
                                ;
006440  020127  000020                  CMP     R1,#FN.QRY              ;QUERY ?
006444  001403                          BEQ     91$                     ;BRANCH IF YES
006446  012701  000002                  MOV     #2,R1                   ;DBU LOG FILE
006452  000402                          BR      92$
006454  012701  000000          91$:    MOV     #0,R1                   ;QUERY LOG FILE
006460  012700  004334'         92$:    MOV     #FDBLOG,R0              ;R0->FDB OF LOG FILE
006464                                  CALL    BLDLOG                  ;BUILD FILENAME BLOCK
006470                                  OFNB$W                          ;OPEN LOG FILE
006502  105760  000052                  TSTB    F.ERR(R0)               ;ERROR ?
006506  003043                          BGT     93$                     ;BRANCH IF NO
006510  005267  175456                  INC     LOGFLG                  ;SHOW DK1 OFFLINE
006514  116004  000052                  MOVB    F.ERR(R0),R4            ;R4 = OPEN ERROR
006520  010467  176462                  MOV     R4,PAR
006524  016667  000004  176456          MOV     4(SP),PAR+2             ;ADD'S OF CALLING ROUTINE
006532                                  MOUT$S  #LOGERR,#PAR
006552  000421                          BR      93$
                                ;
006554  116004  000052          90$:    MOVB    F.ERR(R0),R4            ;YES, GET ERROR VALUE
006560  010467  176422                  MOV     R4,PAR
006564  016667  000004  176416          MOV     4(SP),PAR+2             ;GET ADDR OF CALLING ROUTINE
006572                                  MOUT$S  #OPERR,#PAR             ;ISSUE THE ERROR MESSAGE
006612  000261                          SEC
006614  000401                          BR      99$
006616                          93$:
006616  000241                          CLC
006620                          99$:
```

```
     547 006620  012604                          MOV     (SP)+,R4                ;RESTORE R4
     548 006622  012600                          MOV     (SP)+,R0                ;RESTORE R0
     549 006624  000207                          RETURN
     550                                ;+
     551                                ;
     552                                ; BUILD FILE NAME BLOCKS IN FDB
     553                                ;
     554                                        .MCALL  FDOF$L,FCSBT$
     555 006626                                 FDOF$L
     556 006626                                 FCSBT$
     557                                ;
     558                                ; BLDNFL - BUILD FILE NAME BLOCK FOR NEW FILE
     559                                ;
     560                                ;   INPUT: R0 - FDB ADDRESS
     561                                ;          R1 - FILE NUMBER (FN.XXX)
     562                                ;  OUTPUT: ALL REGISTERS PRESERVED
     563                                ;
     564 006626  062700  000102         BLDLOG: ADD     #F.FNB,R0               ;POINT TO FILE NAME BLOCK
     565 006632  005060  000000                 CLR     N.FID(R0)               ;CLEAR FID
     566 006636  005060  000002                 CLR     N.FID+2(R0)
     567 006642  005060  000004                 CLR     N.FID+4(R0)
     568 006646  010246                         MOV     R2,-(SP)                ;SAVE R2
     569 006650  016102  004532'                MOV     FNINDX(R1),R2           ;FILE NAME/TYPE ADDRESS
     570 006654  011260  000006                 MOV     (R2),N.FNAM(R0)         ;FILE NAME/TYPE IN FDB
     571 006660  016260  000002  000010         MOV     2(R2),N.FNAM+2(R0)
     572 006666  016260  000004  000012         MOV     4(R2),N.FNAM+4(R0)
     573 006674  016260  000006  000014         MOV     6(R2),N.FTYP(R0)
     574 006702  005060  000016                 CLR     N.FVER(R0)              ;VERSION IS ZERO
     575 006706  005060  000020                 CLR     N.STAT(R0)              ;CLEAR STATUS
     576 006712  005060  000022                 CLR     N.NEXT(R0)              ;CLEAR WILD CARD WORD
     577 006716  012702  004600'                MOV     #DIRK74,R2              ;R2-> DIRECTORY FID
     578 006722  011260  000024                 MOV     (R2),N.DID(R0)          ;DIRECTORY FID IN FDB
     579 006726  016260  000002  000026         MOV     2(R2),N.DID+2(R0)
     580 006734  016260  000004  000030         MOV     4(R2),N.DID+4(R0)
     581 006742  016160  004556' 000032         MOV     DVINDX(R1),N.DVNM(R0)   ;DEVICE NAME
     582 006750  016160  004562' 000034         MOV     UNINDX(R1),N.UNIT(R0)   ;DEVICE UNIT
     583 006756  012602                         MOV     (SP)+,R2                ;RESTORE R2
     584 006760  162700  000102                 SUB     #F.FNB,R0               ;RESTORE R0
     585 006764  000207                         RTS     PC                      ;RETURN
     586                                ;+
     587                                ;
     588                                ; **--WRITE ROUTINE
     589                                ;
     590                                ; THIS ROUTINE WRITES A BLOCK OF DATA TO THE SPOOL FILE AND TO THE
     591                                ; EXHANGE LOG FILE.
     592                                ;
     593                                ; INPUTS:
     594                                ;       GENERAL REGISTER USAGE AND R0->THE FDB OF THE SPOOL FILE
     595                                ;
     596                                ; OUTPUTS:
     597                                ;       IF THERE IS AN ERROR THE "C" BIT IS SET, ELSE THE "C"BIT IS
     598                                ;       CLEARED.
     599                                ;
     600                                ;-
     601                                ;
     602 006766  010046                 WRITE:  MOV     R0,-(SP)                ;SAVE R0
     603 006770  010446                         MOV     R4,-(SP)                ;SAVE R4
```

```
 604 006772                                     WRITE$  ,,,,,,,DIRERR          ;WRITE THE BLOCK
 605 007004                                     WAIT$
 606 007010  105767  175146                     TSTB    WRSTST                  ;IS THERE AN ERROR ?
 607 007014  003021                             BGT     90$                     ;NO, GO EXIT
 608 007016  116704  175140          91$:       MOVB    WRSTST,R4               ;
 609 007022  010467  176160                     MOV     R4,PAR
 610 007026  016667  000004  176154             MOV     4(SP),PAR+2
 611 007034                                     MOUT$S  #WRTERR,#PAR
 612 007054  000261                             SEC
 613 007056  000443                             BR      99$
 614                                    ;
 615                                    ;   WRITE TRANSACTION LOG FILE TO DK
 616                                    ;
 617 007060  005767  175106          90$:       TST     LOGFLG                  ;IS DK1 OFFLINE?
 618 007064  001040                             BNE     99$                     ;BRANCH IF YES
 619 007066                                     WRITE$  #FDBLOG,,,,,,DIRERR
 620 007104                                     WAIT$   #FDBLOG
 621 007114  105767  175042                     TSTB    WRSTST
 622 007120  003021                             BGT     92$                     ;BRANCH IF NO ERROR
 623 007122  005267  175044                     INC     LOGFLG                  ;SHOW DK1 OFFLINE
 624 007126  116704  175030                     MOVB    WRSTST,R4               ;R4 = WRITE ERROR
 625 007132  010467  176050                     MOV     R4,PAR
 626 007136  016667  000004  176044             MOV     4(SP),PAR+2             ;ADD'S OF CALLING ROUTINE
 627 007144                                     MOUT$S  #LOGERR,#PAR
 628
 629 007164  000241                  92$:       CLC
 630 007166                          99$:
 631 007166  012604                             MOV     (SP)+,R4
 632 007170  012600                             MOV     (SP)+,R0
 633 007172  000207                             RETURN
 634                                    ;+
 635                                    ;
 636                                    ; CONVERT THE ASCII QUEUE ID TO BINARY.  THE STRING TO BE CONVERTED IS IN
 637                                    ; THE INPUT BUFFER.  A ZERO WORD IS MOVED INTO THE WORD FOLLOWING THE
 638                                    ; 4 ASCII STRING TO TELL THE SYSTEM CONVERSION ROUTINE TO STOP.  AFTER
 639                                    ; CONVERSION THE CLEARED WORD IS RESTORED.
 640                                    ;
 641                                    ; INPUTS:
 642                                    ;       NONE
 643                                    ;
 644                                    ; OUTPUTS:
 645                                    ;       R1 = THE BINARY VALUE OF THE ASCII NUMERIC STRING
 646                                    ;
 647                                    ;-
 648                                    ;
 649 007174                          CONVRT:
 650 007174  010046                             MOV     R0,-(SP)
 651 007176  010246                             MOV     R2,-(SP)
 652 007200  016746  170604                     MOV     INBUF+8.,-(SP)          ;SAVE THE TERMINATOR CHARACTERS
 653 007204  012700  000004'                    MOV     #INBUF+4.,R0            ;R0->START OF THE STRING
 654 007210                                     CALL    $CDTB                   ;CONVERT THE STRING TO BINARY
 655 007214  012667  170570                     MOV     (SP)+,INBUF+8.          ;RESTORE THE TERMINATOR CHARACTERS
 656 007220  012602                             MOV     (SP)+,R2
 657 007222  012600                             MOV     (SP)+,R0
 658 007224  000207                             RETURN
 659                                    ;+
 660                                    ;
```

```
                                ; **--DIRERR ROUTINE
                                ;
                                ; THIS ROUTINE REPORTS DIRECTIVE ERRORS
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       OUTPUT ERROR MESSAGE
                                ;
                                ;-
                                ;
007226                          DIRERR:
007226  016767  000000G 175752          MOV     $DSW,PAR                ;GET THE DIRECTIVE STATUS ERROR
007234  011667  175750                  MOV     (SP),PAR+2              ;GET ADDRESS OF CALLING ROUTINE
007240                                  MOUT$S  #DIR,#PAR               ;OUTPUT ERROR MESSAGE
007260  000207                          RETURN
                                ;+
                                ;
                                ; ERROR REPORTING ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
                                ;
007262                          ERROR:
007262                                  MOUT$S  #IRT                    ;REPORT THE ERROR
007300  000207                          RETURN
                                ;**************
005232'                                 .END    MAIN
```

```
BITVAL=  000000          B.QMAP  000234      010 FDBLOG  004334R       FO.WRT=  000016          IRT     005020R
BIT0  =  000001          B.QSPL  000316      010 FD.BLK=  000010        FUCLEN=  000006          LGQNAM  004536R
BIT1  =  000002          B.QTTM  000076      010 FD.CCL=  000002        F.ACTL=  000076          LGUNAM  004546R
BIT10 =  002000          B.QUQP  000056      010 FD.COM=  020000        F.ALOC=  000040          LOGERR  005126R
BIT11 =  004000          B.SFDB  000010      010 FD.CR =  000002        F.BBFS=  000062          LOGFLG  004172R
BIT12=  010000           B.SIZE  000772      010 FD.DIR=  000010        F.BDB =  000070          LOGNBK  004474R
BIT13 =  020000          B.SNDP  000012      010 FD.FID  000000     003 F.BGBC=  000057          LUN1  =  000001
BIT14 =  040000          B.SSQ   000004      010 FD.FNB  000006     003 F.BKDN=  000026          M     =  000062
BIT15 =  100000          B.SSQF  000050      010 FD.FTN=  000001        F.BKDS=  000020          MAIN    005232R
BIT2  =  000004          B.STAT  000044      010 FD.FVR  000004     003 F.BKEF=  000050          MORE    005412R
BIT3  =  000010          B.STTE  000053      010 FD.F11=  040000        F.BKP1=  000051          MSGOUT=  ******  GX
BIT4  =  000020          B.UDOC  000110      010 FD.INS=  000010        F.BKST=  000024          N     =  000002
BIT5  =  000040          CF.B0 =  000070         FD.ISP=  002000        F.BKVB=  000064          NB.DEV=  000200
BIT6  =  000100          CF.B2 =  000067         FD.LEN  000010     003 F.CHR =  000075          NB.DIR=  000100
BIT7  =  000200          CF.B4 =  000066         FD.MNT=  100000        F.CNTG=  000034          NB.NAM=  000004
BIT8  =  000400          CF.B6 =  000065         FD.OSP=  004000        F.DFNB=  000046          NB.SD1=  000400
BIT9  =  001000          CF.DR0=  000064         FD.PLC=  000004        F.DSPT=  000044          NB.SD2=  001000
BLDLOG  006626R          CF.DR1=  000063         FD.PRN=  000004        F.DVNM=  000134          NB.SNM=  000040
BLDNFL=  ******  GX      CH.AND=  000001         FD.PSE=  010000        F.EFBK=  000010          NB.STP=  000020
BS.CLS=  000002          CONVRT  007174R         FD.RAH=  000001        F.EFN =  000050          NB.SVR=  000010
BS.DBU=  000004          DBSLEN=  000116         FD.RAN=  000002        F.EOBB=  000032          NB.TYP=  000002
BS.INA=  000000          DH.BF0  000002      005 FD.REC=  000001        F.ERR =  000052          NB.VER=  000001
BS.OPN=  000001          DH.BF1  000004      005 FD.RWM=  000001        F.FACC=  000043          N.BFAC=  000004
BS.SRC=  000003          DH.CTL  000000      005 FD.SDI=  000020        F.FFBY=  000014          N.BHGH=  000006
BYTE0 =  000000          DH.DMC  000010      005 FD.SQD=  000040        F.FNAM=  000110          N.BTCH=  000004
BYTE1 =  000001          DH.FLG  000006      005 FD.TTY=  000004        F.FNB =  000102          N.BUFB=  004000
BYTE2 =  000002          DIR     004756R         FD.WBH=  000002        F.FTYP=  000116          N.BUFW=  002000
BYTE3 =  000003          DIRDS1  004566R         FF.CHR=  000005        F.FVER=  000120          N.DID =  000024
BYTE4 =  000004          DIRDT1  004572R         FF.NV =  000003        F.HIBK=  000004          N.DVNM=  000032
BYTE5 =  000005          DIRERR  007226R         FF.POE=  000002        F.LUN =  000042          N.FID =  000000
BYTE6 =  000006          DIRK74  004600R         FF.RWD=  000001        F.MBCT=  000054          N.FNAM=  000006
BYTE7 =  000007          DN.DCK  000000      013 FF.RWF=  000006        F.MBCI=  000055          N.FOS =  000764
BYTE8 =  000010          DN.NTP  000004      013 FF.SPC=  000004        F.MBFG=  000056          N.FTYP=  000014
BYTE9 =  000011          DN.NXT  000006      013 FNINDX  004532R        F.NRBD=  000024          N.FVER=  000016
BYTVAL=  000012          DN.ROT  000002      013 FN.DBR  000026     011 F.NREC=  000030          N.NEXT=  000022
B.BSTA  000054       010 DN.SIZ  000010      013 FN.DBS  000022     011 F.OVBS=  000030          N.PKSZ=  000020
B.CNTX  000046       010 DOC     005642R         FN.DHR  000040     011 F.RACC=  000016          N.PKTS=  000043
B.CQUQ  000060       010 DSPTCH  005570R         FN.EMA  000012     011 F.RATT=  000001          N.QURY=  000031
B.FEMA  000132       010 DVINDX  004556R         FN.EMB  000014     011 F.RCNM=  000034          N.STAT=  000020
B.FEMB  000142       010 EF.1  =  000001         FN.EMC  000016     011 F.RCTL=  000017          N.SUNT=  000002
B.FEMC  000152       010 ERROR   007262R         FN.FSA  000000     011 F.RSIZ=  000002          N.UNIT=  000034
B.FFSA  000202       010 FA.APD=  000100         FN.FSB  000002     011 F.RTYP=  000000          OFFLIN  005650R
B.FFSB  000212       010 FA.CRE=  000010         FN.FSC  000004     011 F.SEQN=  000100          OPEN    006406R
B.FFSC  000222       010 FA.DLK=  001000         FN.LGQ  000034     011 F.SPDV=  000072          OPERR   004652R
B.FMHR  000172       010 FA.ENB=  100000         FN.LGU  000036     011 F.SPUN=  000074          PAR     005206R
B.FQLS  000162       010 FA.EXC=  002000         FN.MFO  000024     011 F.STBK=  000036          PARSE   005530R
B.FSAZ  000100       010 FA.EXT=  000004         FN.MHR  000010     011 F.UNIT=  000136          PAR$$$=  000027
B.FSBZ  000102       010 FA.NSP=  000100         FN.NMB  000044     011 F.URBD=  000020          PURGE   005712R
B.FSCZ  000104       010 FA.POS=  010000         FN.QLS  000006     011 F.VBN =  000064          PUTQUB=  ******  GX
B.HBLK  000120       010 FA.RD =  000001         FN.QRY  000020     011 F.VBSZ=  000060          QE.RQ1=  000144
B.HDOC  000114       010 FA.RWD=  004000         FN.SF0  000030     011 GETFRE=  ******  GX      QHL     006014R
B.HRLP  000126       010 FA.SEQ=  040000         FN.SF1  000032     011 INBUF   000000R         QIOST   004166R
B.HRLR  000122       010 FA.SHR=  000040         FN.SHD  000042     011 INL     005336R         Q.FDSC  000004      007
B.HRLW  000124       010 FA.TMP=  000020         FO.APD=  000106        IOSTAT  004156R         Q.NQBK  000000      007
B.NMBR  000052       010 FA.WCK=  020000         FO.MFY=  000002        IO.INL=  ******  GX     Q.NUHL  000002      007
B.NQRY  000232       010 FA.WRT=  000002         FO.RD =  000001        IO.RLB=  ******  GX     Q.SIZE  000014      007
B.QLSZ  000106       010 FDB     004174R         FO.UPD=  000006        IO.RTC=  ******  GX     RCVERR  004606R
```

SYMBOL TABLE

```
RECEVE  006256R        SR.SUN  000000      002 SU.IDL= 000000        WORD0 = 000000        XHLMER= 000002
R.FIX = 000001         SR.TWS  000056      002 SU.LOD= 000001        WORD1 = 000002        XHOTSK= 000010
R.SEQ = 000003         SR.WSL  000052      002 SU.SRC= 000002        WORD2 = 000004        XMSCHE= 000000
R.VAR = 000002         SR.YR   000004      002 SU.SRR= 000005        WORD3 = 000006        XQTS  = 000003
SR.ARE  000114     002 SR.1IN  000024      002 SU.XPD= 000003        WORD4 = 000010        XQT0  = 000001
SR.ARS  000106     002 SR.1IP  000016      002 S.BFHD= 000020        WORD5 = 000012        XSULOA= 000005
SR.DAY  000010     002 SS.FID  000002      004 S.FATT= 000016        WORD6 = 000014        $CDTB = ****** GX
SR.DLT  000014     002 SS.FNB  000010      004 S.FDB = 000140        WORD7 = 000016        $DSW  = ****** GX
SR.ECB  000047     002 SS.FVR  000006      004 S.FNAM= 000006        WORD8 = 000020        $$$ARG= 000002
SR.ECH  000046     002 SS.LEN  000012      004 S.FNB = 000036        WORD9 = 000022        $$$T1 = 000067
SR.ECL  000050     002 SS.STT  000000      004 S.FNBW= 000017        WRDVAL= 000024        .CLOSE= ****** G
SR.FIB  000012     002 ST.ASZ  000020      006 S.FNTY= 000004        WRITE   006766R       .DLFNB= ****** GX
SR.GRE  000100     002 ST.BSZ  000024      006 S.FTYP= 000002        WRSTST  004162R       .FINIT= ****** G
SR.GRS  000072     002 ST.BTC  000000      006 S.HRL = 000240        WRTERR  004714R       .FSRCB= ****** G
SR.LEN  000122     002 ST.CSZ  000030      006 S.NFEN= 000020        WRT.DK= 000003        .GTDIR= ****** GX
SR.LIN  000066     002 ST.HRL  000010      006 TBL     004000R       WRT.EF= 000002        .OPFNB= ****** G
SR.LIP  000062     002 ST.LEN  000044      006 TBLSZ$= 000012        WRT.LU= 000002        .WAIT = ****** G
SR.MON  000006     002 ST.QRY  000002      006 UNINDX  004562R       XBATCH= 000013        .WRITE= ****** G
SR.NDC  000042     002 ST.QSZ  000034      006 WN.NTP  000004    012 XDBLOA= 000004        ...GBL= 000000
SR.NDS  000036     002 ST.SCH  000040      006 WN.NXT  000006    012 XDBPRO= 000012        ...PC1= 004334R
SR.NIN  000030     002 ST.UHL  000004      006 WN.ROT  000002    012 XDMCIN= 000006        ...PC2= 004530R
SR.NIP  000022     002 ST.XLT  000014      006 WN.SIZ  000010    012 XFOSMR= 000007        ...PC3= 004334R
SR.SDB  000032     002 SU.DBU= 000004          WN.SRC  000000    012 XGTSRE= 000014        ...TPC= 000020
SR.SRC  000002     002 SU.DON= 000006          WN.TYP  000001    012 XHITSK= 000011

. ABS.  000000   000
        007302   001
SRCOFF  000122   002
FDSCOF  000010   003
SUSOFF  000012   004
DHROFF  000012   005
STTOFF  000044   006
QSPLOF  000014   007
BSTOFF  000772   010
FNOFFS  000044   011
WNODOF  000010   012
DNODOF  000010   013
$$FSR1  000000   014
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  7316 WORDS  (29 PAGES)
DYNAMIC MEMORY:  8084 WORDS  ( 31 PAGES)
ELAPSED TIME:  00:01:06
HITSK,HITSK/-SP=[20,1]JP,M,HITSK
```

TABLE OF CONTENTS

10- 2 INTRODUCTION
11- 153 MAIN ROUTINE
12- 179 RCVFIL ROUTINE
13- 220 SPOOLF
14- 264 NOFILE
15- 295 SDHR
16- 318 DSPTCH ROUTINE
17- 354 STATUS ROUTINE
18- 404 LENGTH ROUTINE
19- 436 OPEN ROUTINE
20- 468 SEND ROUTINE
21- 504 READ ROUTINE
22- 537 DIRERR ROUTINE

Approved For Release 2005/07/28 : CIA-RDP85-00514R000100030001-3

```
                        .TITLE  HOTSK -- HOST OUTPUT TASK
                        .SBTTL  INTRODUCTION
                        .IDENT  /01/
                ;+
                ;
                ; H O S T   O U T P U T   T A S K
                ;
                ;
                ; THE PURPOSE OF THIS TASK IS TO SEND DATA (IN-CORE BUFFERS AND SPOOL FILES)
                ; FROM THE MASTER COMPUTER TO THE HOST SYSTEM.  THERE ARE NINE TYPES OF
                ; DATA THAT IS SENT TO THE HOST SYSTEM:
                ;
                ;       1)      BATCH CUT-OFF, SPOOL FILE
                ;       2)      SINGLE BLOCK DOCUMENT HIT REPORTS, IN-CORE
                ;       3)      MULTI-BLOCK DOCUMENT HIT REPORTS, IN-CORE, SPOOL FILE
                ;       4)      FLU OCCURRANCE SUMMARY, SPOOL FILE
                ;       5)      START/STOP QUERIES, IN-CORE
                ;       6)      DATA BASE ACKNOWLEDGEMENTS, SPOOL FILE
                ;       7)      RETRIEVED DOCUMENT, SPOOL FILE
                ;       8)      QUERY ERROR, SPOOL FILE
                ;       9)      START MASS UPDATE, IN-CORE
                ;
                ; BOTH THE IDENTIFIERS OF SPOOL FILES, AND THE COMMAND BYTES OF THE
                ; IN-CORE EXCHANGES ARE SENT GIVEN TO HOTSK VIA THE RECEIVE DATA QUEUE.
                ;
                ; IF AT ANY TIME THERE IS A LINK ERROR THE LINK WILL BE RESTARTED BY THE
                ; HITSK.  ANY DATA THAT IS IN THE PROCESS OF BEING OUTPUT WILL BE RE-QUEUED
                ; AND RE-SENT WHEN THE LINK RETURNS TO SERVICE.  THE EXCEPTION BEING WHEN
                ; A PARITY ERROR IS DETECTED BY THE HOST SYSTEM.  THIS OCCURES BECAUSE
                ; THERE IS NO ACK/NAK PROTOCOL ON THE ACC LINK.  WHEN THE PARITY ERROR
                ; IS DETECTED AND A RESTART IS INVOKED THE IN-CORE BUFFER THAT WAS IN THE
                ; PROCESS OF BEING OUTPUT WILL BE LOST.  THE SPOOL FILES WILL BE CLOSED,
                ; AND RETRANSMITTED WHEN THE LINK RETURNS TO SERVICE.
                ;
                ; DATE WRITTEN:          5 MARCH 1979
                ;
                ; DATE MODIFIED:
                ;
                ;
                ; REGISTER USAGE:
                ;       R0 = ADDRESS OF THE FILE DESCRIPTOR BLOCK (FDB)
                ;       R1 = WORKING REGISTER
                ;       R2 = NO. OF BLOCKS TO BE SENT OVER THE ACC LINK
                ;       R3 = ADDRESS OF THE TABLE ENTRY WHICH RELATES RECORD TYPE
                ;            TO TYPE AND PACKING MODE
                ;       R4 = PACKING MODE
                ;       R5 = ADDRESS OF BUFFER TO BE INPUT OR OUTPUT
                ;
                ;
                ; MACROS
                ;
                        .MCALL  QIOW$S,RCVD$S,DECL$S,SPND$S,CLOSE$
                        .MCALL  QFNB$R,READ$,WAIT$,SETF$S
                        .MCALL  FDBDF$,FDRC$A,FDBK$A,FDOP$A,FSRSZ$
                        .MCALL  ALTP$S
                ;
                .MACRO  ENT,ID,TYPE,PACK,TXTLGT,?L1
```

```
                                        L1:
                                        .WORD   "ID
                                        .WORD   TYPE
                                        .WORD   PACK
                                        .WORD   TXTLGT
                                        VECLN$  =       . - L1
                                .ENDM
                                .MACRO  MSGSTR,STRING,?L1,?L2
                                        .WORD   L2-L1
                                        .WORD   L1
                                L1:
                                        .ASCII  /STRING/
                                L2:
                                        .EVEN
                                .ENDM
                                ;
                                ; EQUATED VARIABLES
                                ;
000001                          AC.LUN  =       1               ;AC HANDLER LUN FOR TRANSMISSION
000001                          AC.EF   =       1               ;AC HANDLER EVENT FLAG
000002                          FIL.LU  =       2               ;SPOOL FILE DISK LUN
000002                          FIL.EF  =       2               ;SPOOL FILE EVENT FLAG
                                ;
                                ; DATA BUFFERS AND DATA STORAGE AREAS
                                ;
                                                                ;REGION DEFINITION BLOCK
000000                          EUF:    .BLKB   N.BUFB          ;SPOOL FILE BUFFER
004000                          PAR:    .BLKW   2.              ;MOUT$S OUTPUT PARAMETER LIST
004004                          FILST:  .BLKW   2.              ;SPOOL FILE I/O STATUS BLOCK
004010                          ACST:   .BLKW   2.              ;ACC I/O STATUS BLOCK
004014                          RCVBUF: .BLKW   15.             ;BUFFER FOR RCVD$S MACRO
004052                          STBUF:  .BLKW   3.              ;BUFFER FOR STATUS FUNCTION WORD THAT IS
                                                                ;SEND TO THE HOST
                                ;
                                ; TABLE THAT GIVES THE RELATIONSHIP BETWEEN RECORD ID AND ITS TYPE AND PACKING
                                ; MODE.
                                ;
004060                          VEC:    ENT     DM,15.,1.2048.  ;MULTI-BLOCK DHR
004070                          DHRI:   ENT     HD,4.,1.2048.   ;SINGLE BLOCK DHR
004100                          ECO:    ENT     CB,3.,1.2048.   ;BATCH CUTOFF
004110                          FLU:    ENT     OF,1.,0.2048.   ;FOS
004120                          SPQ:    ENT     SP,2.,2.32.     ;STOP QUERIES
004130                          STQ:    ENT     ST,2.,2.32.     ;START QUERIES
004140                          IBA:    ENT     AD,10.,4.2046.  ;UPDATE ACK
004150                          RET:    ENT     TR,12.,3.2048.  ;RETRIEVED DOCUMENT
004160                          ERR:    ENT     EQ,19.,1,32.    ;QUERY ERROR
004170                          SMU:    ENT     SM,17.,2,32.    ;START MASS UPDATE
004200                          NUL:    ENT     NL,0,0,0        ;NULL ENTRY TO TELL SOFTWARE OF TABLE END
                                ;
                                ; FILE DESCRIPTOR BLOCK
                                ;
004210                          FDB:    FDBDF$
004350                                  FDRC$A  FD.RWM
004350                                  FDBK$A  BUF,N.BUFB,,FIL.EF,FILST
004350                                  FDOP$A  FIL.LU
004350                                  FSRSZ$  0
                                ;
```

```
                                        ;
                                        ; ERROR MESSAGES
                                        ;
004350                          RCVER:  MSGSTR  <RECEIVE DIRECTIVE ERROR %D>
004406                          IRT:    MSGSTR  <INVALID RECORD TYPE " %2A ">
004446                          STE:    MSGSTR  <STATUS XMIT ERROR %D., GRP IN PROGRESS>
004522                          DER:    MSGSTR  <DETACH DIRECTIVE ERROR %D>
004560                          FLE:    MSGSTR  <FILE LENGTH ERROR>
004606                          FOE:    MSGSTR  <FILE OPEN ERROR %D>
004636                          TE:     MSGSTR  <BLOCK XMIT ERROR %D., GRP IN PROGRESS>
004710                          FRE:    MSGSTR  <FILE READ ERROR %D>
004740                          DIRER:  MSGSTR  <DIRECTIVE STATUS %D, %O>
004776                          CBYTES: MSGSTR  <ILLEGAL COMMAND BYTES %1O, %1O>
                                        ;
                                        ;   EXCHANGE ID PARSE TABLE
                                        ;
005040  006     004             EXCHID: .BYTE   6,4             ;DBACK OR RETRIEVED DOC.
005042  005322'                        .WORD   SPOOLF
005044  006     005                     .BYTE   6,5             ;SINGLE BLOCK DHR
005046  005554'                        .WORD   SDHR
005050  006     006                     .BYTE   6,6             ;MULTI-BLOCK DHR SPOOL FILE
005052  005322'                        .WORD   SPOOLF
005054  007     000                     .BYTE   7,0             ;FOS
005056  005322'                        .WORD   SPOOLF
005060  000     003                     .BYTE   0,3             ;BATCH CUT-OFF MSG
005062  005322'                        .WORD   SPOOLF
005064  000     004                     .BYTE   0,4             ;QUERY ERROR
005066  005322'                        .WORD   SPOOLF
                                        ;
005070  000     000                     .BYTE   0,0             ;START QUERIES
005072  005452'                        .WORD   NOFIL0
005074  000     001                     .BYTE   0,1             ;STOP QUERIES
005076  005470'                        .WORD   NOFIL1
005100  000     002                     .BYTE   0,2             ;START MASS UPDATE
005102  005506'                        .WORD   NOFIL2
005104  000000                        .WORD   0               ;END OF TABLE
005106  000000                        .WORD   0
```

```
                                .SBTTL  MAIN ROUTINE
                        ;+
                        ;
                        ; MAIN ROUTINE
                        ;
                        ; THE PURPOSE OF THIS ROUTINE IS TO GET DATA AND TO SEND IT TO THE HOST
                        ; SYSTEM.  IF THERE IS NO DATA THE TASK WILL SUSPEND AND WAIT FOR ANOTHER
                        ; TASK TO RESUME IT WHEN DATA IS AVAILABLE.
                        ;
                        ; INPUTS:
                        ;       NONE
                        ;
                        ; OUTPUTS:
                        ;       NONE
                        ;
                        ;-
                        ;
                        START:
                        MORE:
                                ALTP$S  ,#150,          ;INCREASE PRIORITY TO 150
                                CALL    RCVFIL          ;SEE IF THERE IS ANYTHING IN THE RCVDAT QUEUE
                                BCS     MORE            ;YES THERE WAS, GO SEE IF THERE IS ANYMORE
                                SPND$S  DIRERR          ;NOTHING IN EITHER THE RBRQ OR RCVDAT QUEUES
                                BR      MORE            ;WHEN THERE IS SOMETHING, ANOTHER TASK WILL
                                                        ;ACITVATE HOTSK
```

```
                                        .SBTTL  RCVFIL ROUTINE
                                ;+
                                ;
                                ; RCVFIL ROUTINE
                                ;
                                ; THIS ROUTINE SENDS EXCHANGES RECEIVED IN THE RCVQ TO THE HOST SYSTEM.
                                ;
                                ;-
                                ;
005152                          RCVFIL:
005152                                  RCVD$S  ,#RCVBUF                ;INPUT DATA BUFFER
005170  005767  000000G                 TST     $DSW                    ;ERROR ON INPUT ?
005174  003012                          BGT     1$                      ;NO, GO PROCESS THE FILE
005176  026727  000000G 000000G         CMP     $DSW,#IE.ITS            ;YES, NO DATA QUEUED ?
005204  001002                          BNE     11$                     ;NO, REPORT THE ERROR
005206  000241                          CLC                             ;YES, TELL CALLER THAT...
005210  000443                          BR      99$                     ;WAS NO DATA
005212                          11$:
005212                                  CALL    DIRERR                  ;OUTPUT ERROR
005216  000241                          CLC
005220  000437                          BR      99$                     ;RETURN
                                ;
                                ;   PARSE COMMAND RECEIVED IN RCVQ
                                ;
005222  012701  005040'        1$:     MOV     #EXCHID,R1              ;R1->PARSE TABLE
005226  022167  176566          4$:     CMP     (R1)+,RCVBUF+4          ;CHECK COMMAND BYTES
005232  001007                          BNE     5$                      ;BRANCH IF NO MATCH
005234                                  ALTP$S
005250  000131                          JMP     @(R1)+                  ;JUMP TO ROUTINE
005252  005721                  5$:     TST     (R1)+                   ;R1->NEXT PARSE TABLE ENTRY
005254  005761  000002                  TST     2(R1)                   ;END OF TABLE?
005260  001362                          BNE     4$                      ;BRANCH IF NO
                                ;
                                ;   ILLEGAL COMMAND BYTES
                                ;
005262  116767  176532  176510          MOVB    RCVBUF+4,PAR            ;PAR = LOW BYTE
005270  116767  176525  176504          MOVB    RCVBUF+5,PAR+2          ;PAR+2 = HI BYTE
005276                                  MOUT$S  #CBYTES,#PAR            ;OUTPUT ERROR MSG
005316  000261                          SEC
005320  000207                  99$:    RETURN
```

```
                                        .SBTTL  SPOOLF
                                ;
                                ;   SUBROUTINE TO TRANSMIT SPOOLED DATA TO HOST
                                ;
                                ; OPEN THE FILE AND COMPUTE THE FILE LENGTH
                                ;
005322  012701  004022'         SPOOLF: MOV     #RCVBUF+5,R1            ;R1->FILE NAME BLOCK
005326                                  CALL    OPEN                    ;OPEN THE SPOOL FILE
005332  103443                          BCS     99$                     ;ERROR, GO EXIT
005334  012705  000000'                 MOV     #BUF,R5                 ;R5->INPUT BUFFER
005340                                  CALL    LENGTH                  ;COMPUTE THE LENGTH
                                ;
                                ; FIND THE RECORD'S TYPE AND PACKING MODE
                                ;       R2 = LENGTH
                                ;
005344                                  CALL    READ                    ;GET THE FIRST BLOCK
005350  103435                          BCS     50$                     ;IF ERROR GO TRY AGAIN
005352                                  CALL    DSPTCH                  ;GET THE TYPE AND PACKING MODE
005356                                  CALL    STATUS                  ;SEND THE STATUS FUNCTION WORDS TO HOST
005362  022767  042101  172410         CMP     #"AD,BUF                ;DBU ACK?
005370  001005                          BNE     1$                      ;BRANCH IF NO
005372  016767  172402  172402         MOV     BUF,BUF+2               ;REVERSE TYPE AND PAD WORDS
005400  005067  172374                  CLR     BUF
005404                          1$:     CALL    SEND                    ;SEND TO THE HOST THE FIRST BLOCK
005410  103415                          BCS     50$                     ;IF ERROR, GO TRY AGAIN
005412  005302                          DEC     R2                      ;ACCOUNT FOR BLOCK JUST SENT
005414  001407                          BEQ     3$                      ;IF THATS ALL GO CLOSE FILE
                                ;
                                ; SEND THE REMAINING BLOCKS IN THE SPOOL FILE
                                ;
005416                          2$:
005416                                  CALL    READ                    ;GET A BLOCK
005422  103407                          BCS     99$                     ;IF ERROR GO EXIT
005424                                  CALL    SEND                    ;SEND A BLOCK
005430  103405                          BCS     50$                     ;IF ERROR GO EXIT
005432  077207                          SOB     R2,2$                   ;GO OUTPUT NEXT BLOCK
005434                          3$:     CALL    .DLFNB                  ;DELTE THE FILE
005440  000261                          SEC                             ;INDICATE THAT A FILE WAS OUTPUT
005442                          99$:
005442  000207                          RETURN
005444                          50$:
005444                                  CLOSE$                          ;AN ERROR WAS DETECTED, CLOSE THE
005450  000724                          BR      SPOOLF                  ;SPOOL FILE AND TRY TO SEND IT AGAIN
```

```
                                        .SBTTL· NOFILE·
                                ;
                                ;   SUBROUTINE·TO·SEND·TO·THE·HOST·A·PREDEFINED·DATA·BLOCK·FOR·THOSE·
                                ;   EXCHANGES·THAT·HAVE·NO·SPOOL·FILE.
                                ;
                                ;   THE·COMMAND·BYTES·INDICATE WHAT·THE·DATA·SHOULD·BE.
                                ;
005452· 016767  176452  172320  NOFIL0: MOV·    STQ,BUF·            ;START·QUERIES·
005460  012767  000001  172314          MOV·    #1,BUF+2·
005466  000414                        BR      NOFILE·
005470  016767  176424  172302· NOFIL1: MOV·    SPQ,BUF·            ;STOP·QUERIES·
005476  012767  000002  172276          MOV·    #2,BUF+2·
005504  000405                        BR      NOFILE·
005506  016767  176456  172264  NOFIL2: MOV·    SMU,BUF·            ;START·MASS·UPDATE·
005514  005067  172262                  CLR·    BUF+2·
                                ;
                                ;
                                ;
005520  012702· 000001          NOFILE: MOV·    #1,R2·              ;R2·=·BLOCK·LENGTH·OF·EXCHANGE
005524  012705  000000'                  MOV·    #BUF,R5             ;R5->OUTPUT·BUFFER·
005530                                  CALL·   DSPTCH·             ;GET·THE·TYPE·AND·PACKING·MODE
005534                                  CALL·   STATUS·             ;SEND·THE·STATUS·FUNCTION·WORDS·
005540  103767                          BCS·    NOFILE·             ;IF·ERROR,·TRY·AGAIN
                                ;
                                ;   SEND·DATA·BLOCK·
                                ;
005542·                                 CALL·   SEND
005546  103764                          BCS·    NOFILE·
005550  000261                          SEC·
005552· 000207                          RETURN·
```

```
                                        .SBTTL  SDHR
                                ;
                                ;   SUBROUTINE TO TRANSMIT TO THE HOST A SINGLE-BLOCK DHR RECEIVED
                                ;   FROM DMCIN VIA DOUBLE IN-CORE BUFFERS.  RCVBUF CONTAINS INFORMATION
                                ;   IDENTIFYING THE BUFFER CONTAINING THE DHR.
                                ;
005554  012702  000001          SDHR:   MOV     #1,R2           ;R2 = BLOCK LENGTH OF EXCHANGE
005560  016705  176240                  MOV     RCVBUF+8.,R5    ;R5->INPUT BUFFER (SAME AS OUTPUT BUFF.)
005564                                  CALL    DSPTCH
005570                                  CALL    STATUS
005574  103767                          BCS     SDHR
                                ;
                                ;   SEND BLOCK
                                ;
005576                                  CALL    SEND
005602  103764                          BCS     SDHR
005604  016701  176212                  MOV     RCVBUF+6,R1     ;R1-> DHR CONTROL TABLE
005610  046761  176212  000000          BIC     RCVBUF+13.,DH.CTL(R1)   ;MARK BUFFER EMPTY
005616                                  SETF$S  DH.FLG(R1)      ;SET EVENT FLAG TO SIGNAL DMCIN
005630                                  DECL$S
005636  000261                          SEC
005640  000207                          RETURN
```

```
                                        .SBTTL  DSPTCH ROUTINE
                                ;+
                                ;
                                ; DSPTCH ROUTINE
                                ;
                                ; THIS ROUTINE SEARCHES THE "VEC" TABLE GIVEN THE RECORD ID.  IF THE
                                ; ID IS FOUND THE RECORD TYPE AND PACKING MODE ARE RETURNED.
                                ;
                                ; INPUTS:
                                ;       R5->BUFFER ADDRESS
                                ;
                                ; OUTPUTS:
                                ;
                                ;       R3-> VEC
                                ;       R4 = PACKING MODE
                                ;
                                ;-
                                ;
005642                          DSPTCH:
005642  012703  004060'                 MOV     #VEC,R3                 ;R3->START OF "VEC" TABLE
005646                          2$:
005646  021315                          CMP     (R3),(R5)               ;TABLE MATCH RECORD ID ?
005650  001421                          BEQ     1$                      ;YES
005652  062703  000010                  ADD     #VECLN$,R3              ;NO, POINT R3 TO THE NEXT ONE
005656  005763  000002                  TST     2(R3)                   ;IS THIS THE END OF THE TABLE ?
005662  001371                          BNE     2$                      ;NO, CONTINUE SEARCHING
005664  011567  176110                  MOV     (R5),PAR                ;YES, OUTPUT ERROR MESSAGE
005670                                  MOUT$S  #IRT,#PAR
005710  000261                          SEC                             ;INDICATE AND ERROR
005712  000403                          BR      99$
005714                          1$:
005714  016304  000004                  MOV     4(R3),R4                ;R4=PACKING MODE
005720  000241                          CLC                             ;INDICATE THAT THE RECORD WAS FOUND
005722                          99$:
005722  000207                          RETURN
```

```
                                        .SBTTL  STATUS ROUTINE
                                ;+
                                ;
                                ; STATUS ROUTINE
                                ;
                                ; SEND A STATUS  FUNCTION WORD TO THE HOST SYSTEM
                                ;
                                ; INPUTS:
                                ;       R2 = BLOCK LENGTH OF EXCHANGE
                                ;       R3->"VEC" TABLE ENTRY WHICH POINTS TO THE RECORD TYPE
                                ;
                                ; OUTPUTS:
                                ;       R3 = BYTE LENGTH OF BLOCK TO BE TRANSMITTED
                                ;       STATUS FUNCTION WORD IS SEND TO THE HOST SYSTEM
                                ;
                                ;-
                                ;
005724                          STATUS:
005724                                  SAVE    R0,R1
                                ;
                                ; PACK THE STATUS WORD WITH THE RECORD TYPE AND LENGTH
                                ;
005730  010201                          MOV     R2,R1                   ;R1=FILE LENGTH
005732  072127  000004                  ASH     #4,R1                   ;ADJUST FOR PUTTING IN THE TYPE
005736  016300  000002                  MOV     2(R3),R0                ;R0=RECORD TYPE
005742  073027  177776                  ASHC    #-2,R0                  ;ADJUST BOTH TYPE AND LENGTH
005746  010067  176100                  MOV     R0,STBUF
005752  010167  176076                  MOV     R1,STBUF+2
005756  016303  000006                  MOV     6(R3),R3                ;R3 = BYTE LENGTH OF BLOCK TO
                                                                        ;    BE TRANSMITTED
005762  005067  176070                  CLR     STBUF+4                 ;CLEAR OPTIONAL DATA
                                ;
                                ; SEND THE STATUS FUNCTION WORD
                                ;
005766                                  QIOW$S  #IO.MOD,#AC,LUN,#AC,EF,,#ACST,,<#STBUF,#6,>,DIRERR
006046  105767  175736                  TSTB    ACST                    ;IS THERE AND I/O ERROR
006052  003021                          BGT     1$                      ;NO
006054  116700  175730                  MOVB    ACST,R0
006060  020027  000000G                 CMP     R0,#IE.DNR              ;DEVICE NOT READY ERROR ?
006064  001412                          BEQ     2$                      ;BRANCH IF YES
006066  010067  175706                  MOV     R0,PAR
006072                                  MOUT$S  #STE,#PAR               ;OUTPUT ERROR MESSAGE
006112  000261                  2$:     SEC                             ;INDICATE ERROR TO CALLER
006114  000401                          BR      99$
006116                          1$:
006116  000241                          CLC                             ;INDICATE SUCCESSFULL XMISSION
006120                          99$:
006120                                  RESTOR  R0,R1
006124  000207                          RETURN
```

```
                                        .SBTTL  LENGTH ROUTINE
                                ;+
                                ;
                                ; THIS ROUTINE COMPUTES THE LENGTH OF A FILE IN 1024 WORD BLOCKS.
                                ;
                                ; INPUTS:
                                ;       R0->FDB
                                ;
                                ; OUTPUTS:
                                ;       R2=LENGTH OF THE FILE IN 1024 WORD BLOCKS
                                ;
                                ;-
                                ;
006126                          LENGTH:
006126                                  SAVE    R3
006130  016002  000010                  MOV     F.EFBK(R0),R2           ;GET HIGH ORDER FCS BLOCK COUNT
006134  016003  000012                  MOV     F.EFBK+2(R0),R3         ;GET LOW ORDER FCS BLOCK COUNT
006140  162703  000001                  SUB     #1,R3                   ;MAKE IT ZERO RELATIVE
006144  005602                          SBC     R2
006146  071227  000004                  DIV     #N.BFAC,R2              ;CONVERT TO HSTS BLOCK COUNT
006152  001404                          BEQ     1$                      ;THERE CANNOT BE ZERO BLOCKS
006154  005703                          TST     R3                      ;THERE MUST BE EVEN MULTIPLES
006156  001002                          BNE     1$                      ;NOT EVEN, REPORT ERROR
006160  000241                          CLC                             ;EVERYTHING IS OK, RETURN
006162  000410                          BR      99$
006164                          1$:
006164                                  MOUT$S  #FLE                    ;REPORT THE ERROR
006202  000261                          SEC                             ;GIVE AN ERROR RETURN
006204                          99$:
006204                                  RESTOR  R3
006206  000207                          RETURN
```

```
                                        .SBTTL  OPEN ROUTINE
                                ;+
                                ;
                                ; OPEN ROUTINE
                                ;
                                ; THE PURPOSE OF THIS ROUTINE IS TO OPEN A SPOOL FILE.
                                ;
                                ; INPUTS:
                                ;       R1->FILE NAME BLOCK THAT IS TO BE OPEN.
                                ;
                                ; OUTPUTS:
                                ;       R0->FDB
                                ;       IF AN ERROR IS DETECTED THE "C" BIT IS SET, ELSE IT IS CLEARED
                                ;
                                ;-
                                ;
000210                          OPEN:
000210  012700  004210'                MOV     #FDB,R0         ;R0->FDB
000214                                  CALL    BLDEFL          ;FILL OUT THE FDB FILE ID SECTION
000220                                  OFNB$R                  ;OPEN THE FILE
000232  105760  000052                  TSTB    F.ERR(R0)       ;IS THERE AN ERROR ?
000236  003016                          BGT     1$              ;NO
000240  116001  000052                  MOVB    F.ERR(R0),R1    ;YES, OUTPUT THE ERROR MESSAGE
000244  010167  175530                  MOV     R1,PAR
000250                                  MOUT$S  #FOE,#PAR
000270  000261                          SEC                     ;INDICATE AN ERROR
000272  000401                          BR      99$
000274                          1$:
000274  000241                          CLC                     ;INDICATE EVERYTHING OK
000276                          99$:
000276  000207                          RETURN
```

```
                                        .SBTTL  SEND ROUTINE
                                ;
                                ;+
                                ;
                                ; SEND ROUTINE
                                ;
                                ; THE PURPOSE OF THIS ROUTINE IS TO SEND A BLOCK OF DATA TO THE HOST SYSTEM.
                                ;
                                ; INPUTS:
                                ;       R5->BUFFER TO BE OUTPUT
                                ;       R4 = PACKING MODE
                                ;       R3 = BYTE LENGTH OF BLOCK TO BE TRANSMITTED TO HOST
                                ;
                                ; OUTPUTS:
                                ;       IF THERE IS AN ERROR THE "C" BIT IS SET, ELSE IT IS CLEARED.
                                ;
                                ;-
                                ;
                                SEND:
                                        SAVE    R5
                                        QIOW$S  #IO.WLB,#AC.LUN,#AC.EF,,#ACST,,<R5,R3,R4>,DIRERR
006356  105767  175426                 TSTB    ACST                    ;IS THERE AN ERROR ?
006362  003021                         BGT     1$                      ;NO
006364  116705  175420                 MOVB    ACST,R5                 ;YES, OUTPUT ERROR MSG
006370  020527  000000G                CMP     R5,#IE.DNR              ;DEVICE NOT READY ERROR?
006374  001412                         BEQ     2$                      ;BRANCH IF YES
006376  010567  175376                 MOV     R5,PAR
006402                                  MOUT$S  #TE,#PAR
006422  000261                 2$:      SEC                             ;INDICATE AN ERROR
006424  000401                         BR      99$
006426                         1$:
006426  000241                         CLC
006430                         99$:
006430                                  RESTOR  R5
006432  000207                         RETURN
```

```
                                        .SBTTL  READ ROUTINE
                                ;+
                                ;
                                ; READ ROUTINE
                                ;
                                ; THE PURPOSE OF THIS ROUTINE IS TO READ ONE 1024 WORD BLOCK FROM A SPOOL FILE.
                                ; THIS ROUTINE ASSUMES THAT THE FILE IS ALREADY OPENED.
                                ;
                                ; INPUTS:
                                ;       R0->FDB
                                ;
                                ; OUTPUTS:
                                ;       IF THERE IS AN ERROR THE "C" BIT IS SET, ELSE IT IS CLEARED.
                                ;
                                ;-
                                ;
006434                          READ:
006434                                  SAVE    R0
006436                                  READ$   ,,,,,,,DIRERR           ;READ A BLOCK
006450                                  WAIT$   ,,,DIRERR               ;WAIT FOR IT TO COMPLETE
006462  105767  175316                  TSTB    FILST                   ;WAS THERE AN ERROR ?
006466  003016                          BGT     1$                      ;NO
006470  116700  175310                  MOVB    FILST,R0                ;YES, OUTPUT ERROR MESSAGE
006474  010067  175300                  MOV     R0,PAR
006500                                  MOUT$S  #FRE,#PAR
006520  000261                          SEC                             ;INDICATE THAT THERE IS AN ERROR
006522  000401                          BR      99$
006524                          1$:
006524  000241                          CLC
006526                          99$:
006526                                  RESTOR  R0
006530  000207                          RETURN
```

```
                                        .SBTTL  DIRERR ROUTINE
                                ;+
                                ;
                                ; DIRERR ROUTINE
                                ;
                                ; THE PURPOSE OF THIS ROUTINE IS TO REPORT DIRECTIVE STATUS ERRORS TO
                                ; THE SYSTEM LOG.
                                ;
                                ; INPUTS:
                                ;       DSW
                                ;
                                ; OUTPUTS:
                                ;       DIRECTIVE STATUS ERROR MESSAGE TO THE SYSTEM LOG
                                ;
                                ;-
                                ;
006532                          DIRERR:
006532  016767  000000G 175240          MOV     $DSW,PAR                ;GET THE DSW ERROR
006540  011667  175236                  MOV     (SP),PAR+2              ;GET ADDR OF CALLING ROUTINE
006544                                  MOUT$S  #DIRER,#PAR             ;OUTPUT THE MESSAGE
006564  000207                          RETURN
                                ;*******
005110'                                 .END    START
```

```
ACST    004010R      B.HRLW  000124      010 FN.EMB  000014      011 F.RTYP=  000000          SMU     004170R
AC.EF=  000001       B.NMBR  000052      010 FN.EMC  000016      011 F.SEQN=  000100          SPOOLF  005322R
AC.LUN= 000001       B.NQRY  000232      010 FN.FSA  000000      011 F.SPDV=  000072          SPQ     004120R
BCO     004100R      B.QLSZ  000106      010 FN.FSB  000002      011 F.SPUN=  000074          SR.ARE  000114      002
BITVAL= 000000       B.QMAP  000234      010 FN.FSC  000004      011 F.STBK=  000036          SR.ARS  000106      002
BIT0    =000001      B.QSPL  000316      010 FN.LGQ  000034      011 F.UNIT=  000136          SR.DAY  000010      002
BIT1    =000002      B.QTTM  000076      010 FN.LGU  000036      011 F.URBD=  000020          SR.DLT  000014      002
BIT10   =002000      B.QUQP  000056      010 FN.MFO  000024      011 F.VBN =  000064          SR.ECB  000047      002
BIT11   =004000      B.SFDB  000010      010 FN.MHR  000010      011 F.VBSZ=  000060          SR.ECH  000046      002
BIT12   =010000      B.SIZE  000772      010 FN.NMB  000044      011 IE.DNR=  ****** GX       SR.ECL  000050      002
BIT13   =020000      B.SNDP  000012      010 FN.QLS  000006      011 IE.ITS=  ****** GX       SR.FIB  000012      002
BIT14   =040000      B.SSQ   000004      010 FN.QRY  000020      011 IO.MOD=  ****** GX       SR.GRE  000100      002
BIT15   =100000      B.SSQF  000050      010 FN.SF0  000030      011 IO.WLB=  ****** GX       SR.GRS  000072      002
BIT2    =000004      B.STAT  000044      010 FN.SF1  000032      011 IRT     004406R          SR.LEN  000122      002
BIT3    =000010      B.STTE  000053      010 FN.SHD  000042      011 LENGTH  006126R          SR.LIN  000066      002
BIT4    =000020      B.UDOC  000110      010 FOE     004606R         M       000062          SR.LIP  000062      002
BIT5    =000040      CBYTES  004776R         FO.RD=  ****** GX       MORE    005110R          SR.MON  000006      002
BIT6    =000100      CF.B0 = 000070          FRE     004710R         MSGOUT= ****** GX        SR.NDC  000042      002
BIT7    =000200      CF.B2 = 000067          F.ACTL= 000076          N       000002          SR.NDS  000036      002
BIT8    =000400      CF.B4 = 000066          F.ALOC= 000040          NOFILE  005520R          SR.NIN  000030      002
BIT9    =001000      CF.B6 = 000065          F.BBFS= 000062          NOFIL0  005452R          SR.NIP  000022      002
BLDEFL= ****** GX    CF.DR0= 000064          F.BDB = 000070          NOFIL1  005470R          SR.SDB  000032      002
BS.CLS= 000002       CF.DR1= 000063          F.BGBC= 000057          NOFIL2  005506R          SR.SRC  000002      002
BS.DBU= 000004       DBA     004140R         F.BKDN= 000026          NUL     004200R          SR.SUN  000000      002
BS.INA= 000000       DBSLEN= 000116          F.BKDS= 000020          N.BFAC= 000004           SR.TWS  000056      002
BS.OPN= 000001       DER     004522R         F.BKEF= 000050          N.BHGH= 000006           SR.USL  000052      002
BS.SRC= 000003       DHRI    004070R         F.BKPI= 000051          N.BTCH= 000004           SR.YR   000004      002
BUF     000000R      DH.BF0  000002      005 F.BKST= 000024          N.BUFB= 004000           SR.1IN  000002      002
BYTE0   =000000      DH.BF1  000004      005 F.BKVB= 000064          N.BUFW= 002000           SR.1IP  000016      002
BYTE1   =000001      DH.CTL  000000      005 F.CHR = 000075          N.DID = 000024           SS.FID  000002      004
BYTE2   =000002      DH.DMC  000010      005 F.CNTG= 000034          N.DVNM= 000032           SS.FNB  000010      004
BYTE3   =000003      DH.FLG  000006      005 F.DFNB= 000046          N.FID = 000000           SS.FVR  000006      004
BYTE4   =000004      DIRER   004740R         F.DSPT= 000044          N.FNAM= 000006           SS.LEN  000012      004
BYTE5   =000005      DIRERR  006532R         F.DVNM= 000134          N.FOS = 000764           SS.STT  000000      004
BYTE6   =000006      DN.DCK  000000      013 F.EFBK= 000010          N.FTYP= 000014           START   005110R
BYTE7   =000007      DN.NTP  000004      013 F.EFN = 000050          N.FVER= 000016           STATUS  005724R
BYTE8   =000010      DN.NXT  000006      013 F.EOBB= 000032          N.NEXT= 000022           STBUF   004052R
BYTE9   =000011      DN.ROT  000002      013 F.ERR = 000052          N.PKSZ= 000020           STE     004446R
BYTVAL= 000012       DN.SIZ  000010      013 F.FACC= 000043          N.PKTS= 000043           STQ     004130R
B.BSTA  000054   010 DSPTCH  005642R         F.FFBY= 000014          N.QURY= 000031           ST.ASZ  000020      006
B.CNTX  000046   010 ERR     004160R         F.FNAM= 000110          N.STAT= 000020           ST.BSZ  000024      006
B.COUQ  000060   010 EXCHID  005040R         F.FNB = 000102          N.SUNT= 000002           ST.BTC  000000      006
B.FEMA  000132   010 FDB     004210R         F.FTYP= 000116          N.UNIT= 000034           ST.CSZ  000030      006
B.FEMB  000142   010 FD.FID  000000      003 F.FVER= 000120          OPEN    006210R          ST.HRL  000010      006
B.FEMC  000152   010 FD.FNB  000006      003 F.HIBK= 000004          PAR     004000R          ST.LEN  000044      006
B.FFSA  000202   010 FD.FVR  000004      003 F.LUN = 000042          QE.RQ1= 000144           ST.ORY  000002      006
B.FFSB  000212   010 FD.LEN  000010      003 F.MBCT= 000054          Q.FDSC  000004       007 ST.OSZ  000034      006
B.FFSC  000222   010 FD.RWM= ****** GX       F.MBC1= 000055          Q.NQBK  000000       007 ST.SCH  000040      006
B.FMHR  000172   010 FILST   004004R         F.MBFG= 000056          Q.NUHL  000002       007 ST.UHL  000004      006
B.FQLS  000162   010 FIL.EF= 000002          F.NRBD= 000024          Q.SIZE  000014       007 ST.XLT  000014      006
B.FSAZ  000100   010 FIL.LU= 000002          F.NREC= 000030          RCVBUF  004014R          SU.DBU= 000004
B.FSBZ  000102   010 FLE     004560R         F.OVBS= 000030          RCVER   004350R          SU.DON= 000006
B.FSCZ  000104   010 FLU     004110R         F.RACC= 000016          RCVFIL  005152R          SU.IDL= 000000
B.HBLK  000120   010 FN.DBR  000026      011 F.RATT= 000001          READ    006434R          SU.LUN= 000001
B.HDOC  000114   010 FN.DBS  000022      011 F.RCNM= 000034          RET     004160R          SU.SRC= 000002
B.HRLP  000126   010 FN.DHR  000040      011 F.RCTL= 000017          SDHR    005554R          SU.SRR= 000005
B.HRLR  000122   010 FN.EMA  000012      011 F.RSIZ= 000002          SEND    006300R          SU.XPD= 000003
```

```
HOTSK -- H    OUTPUT TASK        MA
SYMBOL TABLE

S.BFHD= 000020          VEC     004060R         WORD3 = 000006          XDMCIN= 000006          $$$ARG= 000002
S.FATT= 000016          VECLN$= 000010          WORD4 = 000010          XFOSMR= 000007          $$$T1 = 000067
S.FDB = 000140          WN.NTP  000004      012 WORD5 = 000012          XGTSRE= 000014          .CLOSE= ****** G
S.FNAM= 000006          WN.NXT  000006      012 WORD6 = 000014          XHITSK= 000011          .DLFNB= ****** GX
S.FNB = 000036          WN.ROT  000002      012 WORD7 = 000016          XHLMER= 000002          .FSRCB= ****** G
S.FNBW= 000017          WN.SIZ  000010      012 WORD8 = 000020          XHOTSK= 000010          .OPFNB= ****** G
S.FNTY= 000004          WN.SRC  000000      012 WORD9 = 000022          XMSCHE= 000000          .READ = ****** G
S.FTYP= 000002          WN.TYP  000001      012 WRDVAL= 000024          XQTS  = 000003          .WAIT = ****** G
S.HRL = 000240          WORD0 = 000000          XBATCH= 000013          XQT0  = 000001          ...PC1= 004210R
S.NFEN= 000020          WORD1 = 000002          XDBLOA= 000004          XSULOA= 000005          ...PC2= 004350R
TE      004636R         WORD2 = 000004          XDBPRO= 000012          $DSW  = ****** GX       ...TPC= 000020

. ABS.  000000      000
        006566      001
SRCOFF  000122      002
FDSCOF  000010      003
SUSOFF  000012      004
DHROFF  000012      005
STTOFF  000044      006
QSPLOF  000014      007
BSTOFF  000772      010
FNOFFS  000044      011
WNODOF  000010      012
DNODOF  000010      013
$$FSR1  000000      014
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  6400 WORDS  ( 25 PAGES)
DYNAMIC MEMORY:  7028 WORDS  ( 27 PAGES)
ELAPSED TIME:  00:00:59
HOTSK,HOTSK/-SP=[20,1]P,M,HOTSK
```

TABLE OF CONTENTS

7- 3 INTRO AND MACROS
8- 26 EQUATED SYMBOLS
9- 106 LOCAL DATA
10- 168 ACINI -- ENTRY POINT
11- 265 RSTRT -- SUBROUTINE
12- 349 XMIT -- SUBROUTINE
13- 457 RECEVE -- SUBROUITNE
14- 567 FUNCT -- SUBROUTINE
15- 703 XMTINT ($ACOUT) -- ENTRY POINT
16- 794 RCVINT ($ACINP) -- ENTRY POINT
17- 1005 DMA -- SUBROUTINE
18- 1050 POWER FAIL -- ENTRY POINT
19- 1076 TIME OUT -- ENTRY POINT
20- 1120 CANCEL -- ENTRY POINT
21- 1161 CKERR -- ROUTINE
22- 1208 SETAPR -- ROUTINE
23- 1246 IODONE -- ROUTINE
24- 1288 SETERR -- ROUTINE

```
                                        .TITLE  ACC HANDLER
                                        .IDENT  /01/
                                        .SBTTL  INTRO AND MACROS
                                ;
                                ; ACC HANDLER
                                ;
                                ; DATE WRITTEN:
                                ;       8-FEB-78
                                ;
                                ; DATE MODIFIED:
                                ;
                                ;
                                ; MACROS CALLED
                                        .MCALL  ABODF$,HWDDF$,PKTDF$,TCBDF$,FILIO$,SPCIO$,IOERR$
                                        .MCALL  SCBDF$,UCBDF$,DRERR$
000010                                  SPCIO$          ;DEFINE QIO COMMANDS
                                ;
                                ; LOCAL MACRO DEFINITIONS
                                ;
                                        .MACRO  ENTRY,IO,ROUTINE,RSFLG,BC ;DEFINE ACINI VECTOR TABLE
                                        .WORD   IO/256.
                                        .WORD   ROUTINE
                                        .BYTE   RSFLG,BC
                                        .ENDM
```

```
                                        .SBTTL EQUATED SYMBOLS
                                ;
                                ; EQUATED SYMBOLS
                                ;
000000                                  .PSECT  ACCREG,ABS  ;OFFSETS FOR THE ACC HARDWARE REGISTERS
000000                          R.CSR:  .BLKW   1       ;RECEIVE CHANNEL STATUS REGISTER
000002                          R.DB:   .BLKW   1       ;RECEIVE DATA BUFFER
000004                          R.MA:   .BLKW   1       ;RECEIVE MEMORY ADDRESS
000006                          R.WC:   .BLKW   1       ;RECEIVE WORD COUNT
000010                          T.CSR:  .BLKW   1       ;TRANSMIT CHANNEL STATUS REGISTER
000012                          T.DB:   .BLKW   1       ;TRANSMIT DATA BUFFER
000014                          T.MA:   .BLKW   1       ;TRANSMIT MEMORY ADDRESS
000016                          T.WC:   .BLKW   1       ;TRANSMIT WORD COUNT
000020                          R.EF2:  .BLKW   1       ;RECEIVE EXTERNAL FUNCTION WORD 0 (BITS  0-15)
000022                          R.EF1:  .BLKW   1       ;RECEIVE EXTERANL FUNCTION WORD 1 (     16-31)
000024                          R.EF0:  .BLKW   1       ;RECEIVE EXTERNAL FUNCTION WORD 2 (     32-35)
000026                          T.EF2:  .BLKW   1       ;TRANSMIT EXTERNAL STATUS WORD 0 (BITS  0-15)
000030                          T.EF1:  .BLKW   1       ;TRANSMIT EXTERNAL STATUS WORD 1 (     16-31)
000032                          T.EF0:  .BLKW   1       ;TRANSMIT EXTERNAL STATUS WORD 2 (     32-35)
000034                          MCR:    .BLKW   1       ;MODE CONTROL REGISTER
000036                          MC:     .BLKW   1       ;MAINTENANCE CONTROL
000000                                  .PSECT
                                ;
                                ; RECEIVE CSR AND TRANSMIT CSR REGISTER BIT DEFINITIONS
                                ;
        000001                  DMAGO   =       BIT0    ;RCV AND XMIT DMA GO BIT
        000002                  BYTE    =       BIT1    ;RCV AND XMIT BYTE ADDRESSING MODE
        000004                  BBPDP   =       BIT2    ;RCV PDP BUS BACK
        000010                  BBUNVC  =       BIT3    ;RCV UNIVAC BUS BACK
        000004                  RESET   =       BIT2    ;XMIT, RESET ACC HARDWARE INTERFACE
        000020                  EA16    =       BIT4    ;RCV AND XMIT EXTENDED MEMORY ADDRESS BITS
        000040                  EA17    =       BIT5    ;RCV AND XMIT EXTENDED MEMORY ADDRESS BIT
        000100                  INT     =       BIT6    ;RCV AND XMIT EMABLE INTERRUPT
        000200                  DMADON  =       BIT7    ;RCV AND XMIT DMA DONE
        000400                  RBF     =       BIT8    ;RCV READ BUFFER FULL
        000400                  TBE     =       BIT8    ;XMIT TRANSMIT BUFFER EMPTY
        001000                  WCEZ    =       BIT9    ;RCV AND XMIT, DMA WORD COUNT EQUAL TO ZERO
        004000                  PARITY  =       BIT11   ;RCV, PARITY ERROR ON INPUT
        010000                  DPBF    =       BIT12   ;RCV, DMA INPUT PADDED BECAUSE OF INPUT FUNC
        020000                  UNVCR   =       BIT13   ;RCV AND XMIT, UNIVAC RESET THE CHANNEL
        040000                  NEMR    =       BIT14   ;RCV AND XMIT, NON EXISTANT MEMORY
        100000                  COMP    =       BIT15   ;RCV AND XMIT, COMPOSITE OF BITS 13 AND 14
                                ;
                                ; MODE CONTROL REGISTER
                                ;
        000001                  RPM0    =       BIT0    ;RECEIVE PACKING MODE BIT 0
        000002                  RPM1    =       BIT1    ;RECEIVE PACKING MODE BIT 1
        000004                  RPM2    =       BIT2    ;RECEIVE PACKING MODE BIT 2
        000100                  DA      =       BIT6    ;RCV, ATTENTION UNIVAC IS SENDING DATA
        000200                  FRFUC   =       BIT7    ;RCV, FUNCTION RECEIVED FROM UNIVAC
        000400                  TPM0    =       BIT8    ;XMIT, TRANSMIT PACKING MODE BIT 0
        001000                  TPM1    =       BIT9    ;XMIT, TRANSMIT PACKING MODE BIT 1
        002000                  TPM2    =       BIT10   ;XMIT, TRANSMIT PACKING MODE BIT 2
        100000                  SSTU    =       BIT15   ;XMIT, SEND STATUS TO UNIVAC WITH INTERRUPT
                                ;
                                ; NUMBER OF UNITS
                                ;
```

```
     83          000002                  A$$C11  ==      2       ;TWO UNITS
     84                                  ;
     85                                  ; RESTART REQUEST AND RESTART ACKNOWLEDGEMENT CODES
     86                                  ;
     87          000052                  RSTRTC  =       52      ;RESTART REQUEST
     88          000070                  RACK    =       70      ;RESTART ACKNOWLEDGEMENT
     89                                  ;
     90                                  ; SPECIAL I/O FUNCTION CODES.  THESE ARE ACTUALL A REDEFINITION OF
     91                                  ; EXISTING FUNCTION CODES
     92                                  ;
     93          003000                  IO.TFC  =       IO.MOD          ;TRANSMIT STATUS CODE
     94          003400                  IO.RFC  =       IO.RTC          ;RECEIVE FUNCTION CODE
     95                                  ;
     96                                  ; THE FOLLOWING SYMBOL IS DEFINED FOR THE CALL TO $INTSV WHICH IN
     97                                  ; IN THE INTSV$ MACRO
     98                                  ;
     99          000001                  LD$AC   =       1
    100                                  ;
    101                                  ; DDEFINE RECEIVE AND TRANSMIT "CNTBL" TABLE OFFSETS
    102                                  ;
    103          000000                  RCV     =       0               ;RECEIVE OFFSET
    104          000002                  XMT     =       2               ;TRANSMIT OFFSET
```

```
                                        .SBTTL  LOCAL DATA
                                ;
                                ; LOCAL DATA
                                ;
                                        .IF     GT      ^$$C11-1
000000  000000                  TEMP:   .WORD   0       ;FOR THE INTSV$ MACRO
                                        .ENDC
000002                          CNTBL:  .BLKW   A$$C11  ;LINKAGE BETWEEN FORK PROCESSES AND INTERRUPTS
000006  000000                  NB.UHL: .WORD   0       ;UHL BLOCK COUNT
                                ;
                                ; ACINI DISPATCH TABLE
                                ;
000010                          VEC:    ENTRY   IO.INL,RSTRT,0,0
000016                                  ENTRY   IO.WLB,XMIT,1,1
000024                                  ENTRY   IO.TFC,XMIT,1,0
000032                                  ENTRY   IO.RLB,RECEVE,1,1
000040                                  ENTRY   IO.RFC,RECEVE,1,0
000046                                  ENTRY   0,0,0,0
                                ;
                                ; DRIVER'S DISPATCH TABLE
                                ;
000054                          $ACTBL::
000054  000116'                        .WORD   ACINI   ;DEVICE INITIATOR ENTRY POINT
000056  003052'                        .WORD   CANCEL  ;CANCEL I/O OPERATION ENTRY POINT
000060  002760'                        .WORD   TIMOUT  ;TIME OUT ENTRY POINT
000062  002734'                        .WORD   POWER   ;POWERFAIL ENTRY POINT
                                ;
                                ; RESTART FLAG.  THIS FLAG IS SET WHEN A RESTART (IO.INL) MUST BE
                                ; ISSUED ON THE RECEIVE LUN TO RE-INITIALIZE THE HANDLER AND LINE.
                                ;
000064  000001                  RESTRT: .WORD   1       ;HANDLER IS LOADED IN THE RESTART MODE
                                ;
                                ; THIS VARIABLE CONTAINS THE ADDRESS OF THE UCB WHICH IS REQUESTING
                                ; A RESTART.  AFTER THE RESTART IS COMPLETED THIS ADDRESS IS CLEARED.
                                ;
000066  000000                  RSTADD: .WORD   0
                                ;
                                ; FLAG INDICATES THAT THE ACC HARDWARE IS IN LOOP BACK MODE.
                                ; IN THIS MODE DMA'S DO NOT HAVE TO BE PRECEDED BY A FUNCTION
                                ; OR STATUS WORD.
                                ;
000070  000000                  LOOPB:  .WORD   0
                                ;
                                ;   TABLE OF PACKING MODES INDEXED BY TRANSACTION TYPE
                                ;
000072      000     000     000 PACK:   .BYTE   0,0,0,0,0       ;TRANS. TYPE    =0,1,2,3,4
000075      000     000
000077      003                         .BYTE   3               ;               =5      QRY
000100      003                         .BYTE   3               ;               =6      DOC
000101      004                         .BYTE   4               ;               =7      PURGE
000102      003                         .BYTE   3               ;               =8.     REPLACE
000103      003                         .BYTE   3               ;               =9.     OVERLAY
000104      000                         .BYTE   0               ;               =10.
000105      004                         .BYTE   4               ;               =11.    READ
000106      000                         .BYTE   0               ;               =12.
000107      003                         .BYTE   3               ;               =13.    SUB-DOC
000110      004                         .BYTE   4               ;               =14.    DEL SUB-DOC
```

```
162 000111  000                 .BYTE  0          ;          =. 15.
163 000112  002                 .BYTE  2          ;          =. 16.   REQ: MASS UPDATE
164 000113  000                 .BYTE  0          ;          =. 17.
165 000114  002                 .BYTE  2          ;          =. 18.   END MASS UPDATE
166                             .EVEN
```

```
                                .SBTTL  ACINI -- ENTRY POINT
                        ;+
                        ; **-ACINI-ACC DRIVER INITIATOR
                        ;
                        ; THIS ROUTINE IS ENTERED FROM THE QUEUE I/O DIRECTIVE WHEN AN I/O
                        ; REQUEST IS QUEUED AND AT THE END OF A PREVIOUS I/O OPERATION TO
                        ; PROAGATE THE EXECUTION OF THE DRIVER.  IF THE SPECIFIED CONTROLLER
                        ; IS NOT BUSY, THEN AN ATTEMPT IS MADE TO DEQUEUE THE NEXT I/O
                        ; REQUEST.  ELSE A RETURN TO THE CALLER IS EXECUTED.  IF THE DEQUEUE
                        ; ATTEMPT IS SUCCESSFUL, THEN THE NEXT I/O OPERATION IS INITIATED.
                        ; A RETURN TO THE CALLER IS THEN EXECUTED.
                        ;
                        ; INPUTS:
                        ;
                        ; R5=ADDRESS OF THE UCB OF THE CONTROLLER TO BE INITIATED
                        ;
                        ; OUTPUTS:
                        ;
                        ; IF THE SPECIFIED CONTROLLER IS NOT BUSY AND AN I/O REQUEST IS WAITING
                        ; TO BE PROCESSED, THEN THE REQUEST IS DEQUEUED AND THE I/O OPERATION IS
                        ; INITIATED.
                        ;-
                        ;
000116                  ACINI:
000116                          CALL    $GTPKT          ;GET AN I/O PACKET TO PROCESS
000122  103453                 BCS     RETTE           ;IF C SET CONTROLLER BUSY OR NO REQUEST
                        ;
                        ; THE FOLLOWING ARGUMENTS ARE RETURNED BY $GTPKT
                        ;
                        ;       R1=ADDRESS OF THE I/O REQUEST PACKET
                        ;       R2=PHYSICAL UNIT NUMBER OF THE REQUEST UCB
                        ;       R3=CONTROLLER INDEX
                        ;       R4=ADDRESS OF THE SCB
                        ;       R5=ADDRESS OF THE UCB
                        ;
                        ;
                        ;
                        ; SAVE THE UCB ADDRESS FOR THE INTERRUPT ROUTINE.
                        ;
000124  010563  000002' MOV     R5,CNTBL(R3)
                        ;
                        ; USE THE DISPATCH TABLE TO FIND OUT WHICH ROUTINE TO CALL
                        ;
000130                  1$:
000130  012700  000010' MOV     #VEC,R0         ;R0->START OF DISPATCH TABLE
000134                  5$:
000134  126110  000001G CMPB    I.FCN+1(R1),(R0);DOES INPUT FUNC MATCH TABLE ENTRY ?
000140  001404          BEQ     2$              ;YES, SEE IF ROUTINE CAN BE CALLED
000142  005720          TST     (R0)+           ;IS THIS THE LAST ENTRY IN THE TABLE ?
000144  001435          BEQ     4$              ;YES, INVALID FUNC CODE
000146  022020          CMP     (R0)+,(R0)+     ;NO, GO GET NEXT ENTRY
000150  000771          BR      5$
                        ;
                        ; IF THE RESTART FLAG IS SET FOR THE ENTRY AND THE "RESTRT" FLAG IS
                        ; SET, DO NOT CALL THE ROUTINE, ELSE CALL THE ROUTINE.
                        ;
000152                  2$:
```

```
                        TSTB    4(R0)           ;IS THE RESTART FLAG SET FOR THE ENTRY ?
                        BEQ     3$              ;YES, CALL RESTART
                        TST     LOOPB           ;IF IN LOOP BACK MODE ANYTHING IS LEGIT
                        BNE     3$              ;FLAG SET, SO GO ISSUE I/O
                        TST     RESTRT          ;IS THE RESTART FLAG SET ?
                        BNE     7$              ;YES, RETURN ERROR
                        TSTB    5(R0)           ;IS THE MUST BE ZERO BLOCK COUNT FLAG SET ?
                        BNE     6$              ;NO, TRANSFER FUNCTION MUST HAVE +VE BLOCK COUNT
                        TST     U.CW3(R5)       ;YES, IS THE BLOCK COUNT ZERO ?
                        BEQ     3$              ;YES, GO CALL FUNCTION
                        BR      7$              ;NO, RETURN AN ERROR
                6$:
                        TST     U.CW3(R5)       ;IS THE BLOCK COUNT ZERO  FOR DMA OPERATIONS ?
                        BGT     3$              ;NO, GO DO THE DMA OPERATION
                7$:                             ;REPORT ERROR
                        MOV     #IE.DNR,R0      ;SYSTEM IN RESTART AND WILL NOT ACCEPT
                                                ;THIS FUNCTION, SO NOTIFY THE CALLING
                                                ;TASK
                        CALL    $IODON
                        BR      ACINI           ;GO SEE IF THERE ARE MORE I/O PACKETS
                ;
                ; A VALID FUNCTION CODE WAS FOUND AND IT PASSED THE PROPER RESTART
                ; REQUIREMENTS, SO CALL THE ROUTINE.
                ;
                3$:
                        JSR     PC,@2(R0)       ;CALL THE ROUTINE INDIRECTLY FROM THE
                                                ;TABLE
                        BR      ACINI           ;GO GET ANOTHER I/O PACKET
                ;
                ; AN INVALID FUNCTION WAS FOUND, NOTIFY THE ERRING TASK
                ;
                4$:
                        MOV     #IE.IFC,R0
                        CALL    $IODON
                        BR      ACINI           ;GO GET ANOTHER I/O PACKET
                ;
                ; RETURN TO THE EXECUTIVE
                ;
                RETTE:  RETURN
```

```
                                        .SBTTL  RSTRT -- SUBROUTINE
                                ;+
                                ; **--RSTRT - RESTART ROUTINE.
                                ;
                                ; THIS ROUTINE RESTARTS THE ACC CHANNEL.  THIS ROUTINE IS CALLED WHEN
                                ; A TASK REQUESTS A RESTART VIA AN IO.INL.  THIS REQUEST MUST BE ON THE
                                ; RECEIVE UNIT (UNIT 0).  THE ROUTINE RESETS THE ACC HARDWARE, CLEARS ALL
                                ; THE  ACC HARDWARE REGISTERS, SETS UP THE LOOP BACK MODE (IF NECESSARY)
                                ; SENDS THE RESTART CODE "RSTRTC" AND ENABLES THE INTERRUPTS.
                                ;
                                ; INPUTS:
                                ;       R1=ADDRESS OF THE I/O REQUEST PACKET
                                ;       R2=PHYSICAL UNIT NUMBER OF THE REQUEST UCB
                                ;       R3=CONTROLLER INDEX
                                ;       R4=ADDRESS OF THE SCB
                                ;       R5=ADDRESS OF THE UCB
                                ;
                                ;
                                ; OUTPUTS:
                                ;       THE SAME AS THE INPUTS EXCEPT R0 POINTS TO THE FIRST
                                ;       CSR IN THE ACC HARDWARE REGISTER SET.
                                ;
                                ;-
                                ;
000254                          RSTRT:
                                ;
                                ; IS THE I/O REQUEST ON UNIT 0?  IF SO, CONTINUE, ELSE
                                ; SEND ERROR BACK TO THE REQUESTING TASK AND RETURN TO
                                ; THE CALLING ROUTINE.
                                ;
000254  105765  000000G         TSTB    U.UNIT(R5)              ;IS THE UNIT = 0 ?
000260  001405                  BEQ     3$                      ;YES CONTINUE
000262  012700  000000G         MOV     #IE.LNL,R0              ;NO, GIVE BAD STATUS BACK TO TASK
000266                          CALL    $IODON
000272  000473                  BR      4$                      ;EXIT
000274                          3$:
000274  016400  000000G         MOV     S.CSR(R4),R0            ;R0->FIRST ACC CSR
000300  052760  000004  000010  BIS     #RESET,^.CSR(R0)        ;RESET THE ACC HARDWARE
000306  052760  000004  000010  BIS     #RESET,^.CSR(R0)        ;KEEP DOING IT UNTIL IT SEES IT
000314  052760  000004  000010  BIS     #RESET,^.CSR(R0)        ;DITTO
000322  105761  000000G         TSTB    I.FCN(R1)               ;LOOPBACK REQUESTED ?
000326  001413                  BEQ     1$                      ;NO
                                ;
                                ; SET UP LOOP BACK
                                ;
000330  005267  177534          INC     LOOPB                   ;SHOW WE ARE IN LOOP BACK MODE
000334  116103  000000G         MOVB    I.FCN(R1),R3            ;GET MODIFYER BYTE
000340  042703  177770          BIC     #177770,R3              ;MASK OFF GARBAGE
000344  072327  000002          ASH     #2,R3                   ;SHIFT UP TO PROPER BITS
000350  050360  000000          BIS     R3,R.CSR(R0)            ;SET THE PROPER LOOP BACK MODE
000354  000402                  BR      5$
000356                          1$:
000356  005067  177506          CLR     LOOPB                   ;SHOW THAT WE ARE NOT IN LOOP BACK
000362                          5$:
                                ;
                                ; SHOW THAT WE ARE IN THE RESTART MODE
                                ;
```

```
000362  005065  000000G                CLR     U.CW3(R5)               ;CLEAR RECEIVE BLOCK COUNT
000366  005267  177472                 INC     RESTRT
                                ;
                                ; LOAD THE RESTART CODE INTO THE XMIT STATUS REGISTERS.
                                ;
000372                                  SAVE    R2,R3
000376  012702  000052                  MOV     #RSTRTC,R2              ;R2 = RESTART CODE (52)
000402  073227  177776                  ASHC    #-2,R2                  ;SHIFT RIGTHMOST 2 BITS TO R3
000406  010260  000032                  MOV     R2,T.EF0(R0)            ;LOAD BITS 32-35
000412  010360  000030                  MOV     R3,T.EF1(R0)            ;LOAD BITS 16-31
000416  005060  000026                  CLR     T.EF2(R0)               ;LOAD BITS 0-15
000422                                  RESTOR  R2,R3
000426  052760  100000  000034          BIS     #SSTU.MCR(R0)           ;SET THE GO BIT FOR THE WORD
000434  010567  177426                  MOV     R5,RSTADD               ;SAVE ADDR OF UCB RSTRT ROST
                                ;
                                ; SET THE TIME OUT FOR THE RECEPTION OF THE RESTART ACKNOWLEDGEMENT
                                ; AND ENABLE INTERRUPTS
                                ;
000440  116464  000000G 000000G         MOVB    S.ITM(R4),S.CTM(R4)     ;SET THE TIME OUT
000446  052760  000100  000010          BIS     #INT.T,CSR(R0)          ;ENABLE XMIT INTERRUPTS
000454  052760  000100  000000          BIS     #INT.R,CSR(R0)          ;ENABLE RCV INTERRUPTS
                                ;
                                ; RETURN TO THE CALLING ROUTINE
                                ;
000462                          4$:
000462                                  RETURN
```

```
                                        .SBTTL  XMIT  -- SUBROUTINE
                                ;+
                                ;
                                ; **--XMIT - XMIT ROUTINE
                                ;
                                ; THIS ROUTINE SETS UP TRAMSMIT FUNCTIONS.  THIS CAN INCLUDE BOTH
                                ; DMA AND SEND STATUS OPERATIONS.  THE COMPLETION WILL BE HANDLED
                                ; IN THE XMIT ISR (XMTINT).
                                ; THE ROUTINE FIRST CHECK TO SEE THAT THE TRANSMIT OPERATION IS ON
                                ; UNIT 1.  IF NOT AN ERROR IS RETURNED TO THE REQUESTING TASK.
                                ; THE NUMBER OF DMA BLOCKS THAT IS TO BE OUTPUT IS NEXT CHECKED.
                                ; IF THIS IS NON ZERO FOR A SEND STATUS OPERATION, AN ERROR IS
                                ; RETURNED TO THE CALLING TASK.  ALSO, IF THE NUMBER OF BLOCKS IS ZERO
                                ; AND IT IS A DMA OPERATION, AN ERROR IS RETURNED TO THE CALLING
                                ; TASK.  ELSE, EITHER THE DMA OR SEND STATUS OPERATION IS SETUP.
                                ;
                                ; INPUTS:
                                ;       R1=ADDRESS OF THE I/O REQUEST PACKET
                                ;       R2=PHYSICAL UNIT NUMBER OF THE REQUEST UCB
                                ;       R3=CONTROLLER INDEX
                                ;       R4=ADDRESS OF THE SCB
                                ;       R5=ADDRESS OF THE UCB
                                ;
                                ; REGISTER USAGE:
                                ;
                                ;       R0 IS USED FOR SETTING UP DATA
                                ;       R3->ADDRESS OF THE FIRST ACC HARDWARE REGISTER
                                ;
                                ; OUTPUTS:
                                ;       R0 AND R3 ARE DESTROYED
                                ;
                                ;-
                                ;
000464                          XMIT:
                                ;
                                ; IS THIS THE CORRECT UNIT ?.  IT MUST BE UNIT 1 FOR XMIT OPERATIONS.
                                ;
000464  126527  000000G 000001          CMPB    U.UNIT(R5),#1
000472  001405                         BEQ     3$
000474  012700  !00000G                 MOV     #IE.ILU,R0
000500                                  CALL    IODONE
000504  000514                         BR      99$                     ;EXIT
000506                          3$:
                                ;
                                ; IS THIS A DMA OPERATION ?.
                                ;
000506  126127  000001G 000000C        CMPB    I.FCN+1(R1),#IO.WLB/256, ;WRITE ?
000514  001055                         BNE     1$                      ;NO, IT MUST BE SEND STATUS
                                ;
                                ; SETUP A DMA OPERATION
                                ;
000516  016403  000000G                 MOV     S.CSR(R4),R3            ;R3->START OF ACC REGISTERS
000522  016563  000002G 000014         MOV     U.BUF+2(R5),T.MA(R3)    ;GET LOWER 16 BIT DMA ADDRESS
000530  016500  000000G                 MOV     U.BUF(R5),R0            ;SETUP EXTENSION BITS
000534  042700  177717                  BIC     #177717,R0              ;MASK OFF OTHER BITS
000540  042763  000060  000010         BIC     #60,T.CSR(R3)           ;CLEAR EXTENSION BITS
000546  050063  000010                 BIS     R0,T.CSR(R3)            ;PUT EXTENSION BITS INTO XMIT CSR
```

```
000552  016100  000004G          MOV     I.PRM+4(R1),R0          ;GET XMIT BYTE COUNT
000556  032700  000001          BIT     #BIT0,R0                ;ODD BYTE COUNT?
000562                          BOFF    50$                     ;NO, CONTINUE
000564  012700  000000G          MOV     #IE.BAD,R0              ;YES, REPORT BAD PARAMETER
000570                          CALL    IODONE
000574  000460                  BR      99$                     ;AND EXIT
000576  006200              50$:ASR     R0                      ;CONVERT BYTES TO WORDS
000600  010063  000016          MOV     R0,T.WC(R3)             ;PUT TRANSMIT WORD COUNT
000604  016100  0f9006G          MOV     I.PRM+6(R1),R0          ;GET THE PACKING MODE
000610  042700  177770          BIC     #177770,R0              ;MASK OFF ANY GARBAGE
000614  072027  000010          ASH     #8.,R0                  ;SHIFT FOR MCR
000620  042763  003400  000034  BIC     #3400,MCR(R3)           ;CLEAR BITS 8,9 AND 10
000626  050063  000034          BIS     R0,MCR(R3)              ;SET INTO MCR (BITS 8,9 AND 10)
000632  052763  000001  000010  BIS     #DMAGO,T.CSR(R3)        ;TELL HARDWAR TO SEND DATA
000640  116464  000000G 000000G MOVB    S.ITM(R4),S.CTM(R4)     ;SET THE TIME OUT
000646  000433                  BR      99$                     ;GO EXIT
                            ;
                            ; OUTPUT A 36 BIT STATUS WORD TO THE UNIVAC
                            ;
000650                      1$:
000650                          CALL    SETAPR                  ;SETUP THE APR TO MAP INTO
                                                                ;THE TASKS MEMORY AREA
000654  01640C  000000G          MOV     S.CSR(R4),R3            ;R3->FIRST ACC REGISTER
000660  012063  000032          MOV     (R0)+,T.EF0(R3)         ;GET FIRST WORD OF STATUS
000664  012063  000030          MOV     (R0)+,T.EF1(R3)         ;GET SECOND WORD OF STAIUS
000670  011063  000026          MOV     (R0),T.EF2(R3)          ;GET THRID WORD OF STATUS
000674  012667  000000G          MOV     (SP)+,KISAR6            ;RESTORE APR 6
                            ;
                            ; GET THE BLOCK COUNT FROM THE 36 BIT FUNCTION
                            ;
000700  016300  000030          MOV     T.EF1(R3),R0
000704  042700  140000          BIC     #140000,R0              ;MASK OFF OTHER INFO
000710  072027  177776          ASH     #-2,R0                  ;RIGHT JUSTIFY
000714  010065  000000G          MOV     R0,U.CW3(R5)            ;STORE IN WORD COUNT AREA
                            ;
                            ; SETUP THE TIME OUT
                            ;
000720  116464  000000G 000000G MOVB    S.ITM(R4),S.CTM(R4)
                            ;
                            ; TELL HARDWARE TO SEND THE STATUS
                            ;
000726  052763  100000  000034  BIS     #SSTU,MCR(R3)
000734  000400                  BR      99$                     ;EXIT
                            ;
                            ; RETURN TO CALLING ROUTINE
                            ;
000736                      99$:
000736                          RETURN
                            ;
```

```
                                        .SBTTL  RECEVE -- SUBROUITNE
                                ;+
                                ;
                                ; **--RECEVE - PROCESS A NEW RECEIVE I/O PACKET
                                ;
                                ; THE PURPOSE OF THE RECEVE ROUTINE IS TO SETUP THE ACC
                                ; HARDWARE TO RECEIVE EITHER A 36 BIT EXTERNAL FUNCTION OR
                                ; BLOCK OF WORDS (16 BIT) VIA THE RECEIVE DMA.  IF A REQUEST
                                ; TO RECEIVE A FUNCTION IS ISSUED WHILE THE HANDLER IS IN THE
                                ; PROCESS OF INPUTTING ONE OR MORE DMA BLOCKS, THE REQUEST IS
                                ; REJECTED.  IF AN APPLICATION TASKS TRYS TO ISSUE A DMA
                                ; OPERATION BEFORE THE FUNCTION REQUEST  IS ISSUED, THE DMA
                                ; OPERATION IS REJECTED.
                                ;
                                ; SINCE AN EXTERNAL FUNCTION CAN BE RECEIVED BEFORE THE EXECUTIVE
                                ; GIVES THE I/O PACKET TO THE HANDLER, THE RECEVE
                                ; ROUTINE WILL CHECK TO SEE IF A FUCNTION HAS BEEN INPUT.  IF
                                ; ONE HAS BEEN INPUT, IT IS IMMEDIDATELY RETURNED TO THE
                                ; REQUESTING TASK.  IF THE INPUT FUNCTION IS A RESTART REQUEST
                                ; FROM THE UNIVAC, AN ERROR IS RETURNED TO THE APPLICATION
                                ; TASK.  THEN THE TRANSMIT ROUTINE IS NOTIFIED OF A RESTART.
                                ;
                                ;
                                ; INPUTS:
                                ;       R1=ADDRESS OF THE I/O REQUEST PACKET
                                ;       R2=PHYSICAL UNIT NUMBER OF THE REQUEST UCB
                                ;       R3=CONTROLLER INDEX
                                ;       R4=ADDRESS OF THE SCB
                                ;       R5=ADDRESS OF THE UCB
                                ;
                                ;
                                ;
                                ; REGISTER USAGE:
                                ;
                                ;       R0 IS USED FOR SETTING UP DATA
                                ;       R3->FRIST ACC HARDWARE ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0 AND R3 ARE DESTROYED
                                ;
                                ;-
                                ;
000740                          RECEVE:
                                ;
                                ; FIRST CHECK TO SEE IF THIS IS UNIT 0.  IF NOT RETURN AN ERROR
                                ;
000740  105765  000000G          TSTB    U.UNIT(R5)
000744  001405                  BEQ     1$                      ;NO ERROR
000746  012700  000000G          MOV     #IE.ILU,R0              ;YES THERE IS AN ERROR
000752                          CALL    IODONE                  ;RETURN ERROR TO TASK
000756  000463                  BR      99$                     ;GO RETURN
000760                          1$:
                                ;
                                ; CHECK TO SEE IF A DMA OPERATION IS REQUESTED
                                ;
000760  016403  000000G          MOV     S.CSR(R4),R3            ;R3->1ST ACC HARDWARE REGISTER
000764  126127  000001G 000000C  CMPB    I.FCN+1(R1),#IO.RLB/256.;IS THIS A DMA OP REQUEST ?
```

```
514 000772  001040                          BNE     2$                      ;NO
515                                 ;       BIT     #FRFUC,MCR(R3)          ;YES, FUNCTION INPUT ?
516                                 ;       BEQ     3$                      ;NO
517                                 ;       BIC     #FRFUC,MCR(R3)          ;YES, REPORT ERROR
518                                 ;       MOV     #IE.CNR,R0
519                                 ;       CALL    IODONE
520                                 ;       BR      50$                     ;GO FINISH ERROR REPORTING
521 000774                          3$:
522                                 ;
523                                 ; SET UP FOR DMA INPUT
524                                 ;
525 000774                          4$:
526 000774  016563  000002G 000004          MOV     U.BUF+2(R5),R.MA(R3)    ;GET LOWER 16 BIT ADDRESS
527 001002  016500  000000G                 MOV     U.BUF(R5),R0            ;GET UPPER 2 BITS OF 18 BITS
528 001006  042700  177717                  BIC     #177717,R0              ;MASK OFF GARBAGE
529 001012  042763  000060  000000          BIC     #60,R.CSR(R3)           ;CLEAR EXTENSION BITS
530 001020  050063  000000                  BIS     R0,R.CSR(R3)            ;PLACE INTO EXTENSION BITS
531                                 ;       MOV     I.PRM+6(R1),R0          ;GET PACKING MODE
532                                 ;       BIC     #177770,R0              ;MASK OFF GARBAGE
533                                 ;       BIC     #7,MCR(R3)              ;CLEAR BITS 0, 1 AND 2
534                                 ;       BIS     R0,MCR(R3)              ;PUT INTO PACKING MODE BITS
535 001024  016100  000004G                 MOV     I.PRM+4(R1),R0          ;GET THE INPUT BYTE COUNT
536 001030  032700  000001                  BIT     #BIT0,R0                ;ODD BYTE COUNT?
537 001034                                  BOFF    40$                     ;NO CONTINUE
538 001036  012700  000000G                 MOV     #IE.BAD,R0              ;YES, REPORT BAD PARAMETER
539 001042                                  CALL    IODONE
540 001046  000415                          BR      50$
541 001050  006200                  40$:    ASR     R0                      ;CONVERT BYTES TO WORDS
542 001052  010063  000006                  MOV     R0,R.WC(R3)             ;PUT RECEIVE WORD COUNT
543 001056  052763  000001  000000          BIS     #DMAGO,R.CSR(R3)        ;ACITVATE THE DMA
544 001064  116464  000000G 000000G         MOVB    S.ITM(R4),S.CTM(R4)     ;SET THE TIME OUT
545 001072  000415                          BR      99$                     ;GO EXIT
546                                 ;
547                                 ; PROCESS FUNCTION REQUESTS
548                                 ;
549 001074                          2$:
550 001074                                  CALL    FUNCT
551 001100  000412                          BR      99$                     ;GO EXIT
552                                 ;
553                                 ; FINISH PROCESSING ERRORS BY SETTING THE RESTART FLAG AND INDUCING
554                                 ; AN ERROR INTERRUPT IN THE TRANSMIT SIDE.
555                                 ;
556 001102                          50$:
557 001102  042763  000100  000010          BIC     #INT.T,CSR(R3)          ;DISABLE TRANSMIT AND RECIEVE
558 001110  042763  000100  000000          BIC     #INT.R,CSR(R3)          ;....INTERRUPTS
559 001116  012700  000002                  MOV     #XMT,R0                 ;TELL XMIT THAT ERROR WAS DETECTED
560 001122                                  CALL    SETERR
561                                 ;
562                                 ; RETURN TO THE CALLING ROUTINE
563                                 ;
564 001126                          99$:
565 001126                                  RETURN
```

```
                                        .SBTTL  FUNCT -- SUBROUTINE
                                ;+
                                ;
                                ; **--FUNCT - PROCESS NEW RECEIVE FUNCTION REQUESTS
                                ;
                                ; THIS ROUTINE TAKES A NEW I/O REQUEST FOR AN INPUT FUNCTION
                                ; AND PERFORMS ONE OF THE FOLLOWING ACTIONS:
                                ;
                                ;       1) NOTHING.  THE RECEIVE INTERRUPT WILL RETURN THE
                                ;          FUNCTION WHEN IT IS INPUT.
                                ;
                                ;       2) IF A FUNCTION IS ALREADY WAITING, IT WILL
                                ;          IMMEDIATELY BE RETURNED TO THE APPLICATION TASK.
                                ;
                                ;       3) IF A FUNCTION IS WAITING AND IT IS A RESTART
                                ;          REQUEST, AN ERROR WILL BE RETURNED TO THE
                                ;          APPLICATION TASK AND AN ERROR INTERRUPT
                                ;          GENERATED FOR THE XMIT SIDE.
                                ;
                                ; INPUTS:
                                ;       R1=ADDRESS OF THE I/O REQUEST PACKET
                                ;       R2=PHYSICAL UNIT NUMBER OF THE REQUEST UCB
                                ;       R3=CONTROLLER INDEX
                                ;       R4=ADDRESS OF THE SCB
                                ;       R5=ADDRESS OF THE UCB
                                ;
                                ;
                                ;
                                ; REGISTER USAGE:
                                ;
                                ;       R0 IS USED FOR SETTING UP DATA
                                ;       R3->THE FIRST ACC HARDWARE ADDRESS
                                ;
                                ; OUTPUTS:
                                ;
                                ;       R0 AND R3 ARE DESTROYED
                                ;
                                ;-
                                ;
001130                          FUNCT:
001130  016403  000000G         1$:     MOV     S.CSR(R4),R3            ;R3->FIRST ACC HARDWARE ADDRESS
001134                                  ;
                                ; WAS AN EMERGENCY FUNCTION SENT BY THE UNIVAC
                                ;
001134  032763  010000  000000          BIT     #DPBF,R.CSR(R3)         ;FUNCTION INPUT OVER A DMA ?
001142  001410                          BEQ     2$                      ;NO
001144  012700  000000G                 MOV     #IE.CNR,R0              ;YES, RETURN ERROR
001150                                  CALL    IODONE
001154  042763  010000  000000          BIC     #DPBF,R.CSR(R3)         ;CLEAR THE BIT
001162  000525                          BR      50$                     ;GO PROCESS THE REST OF THE ERROR
001164                          2$:
                                ;
                                ; IS A FUNCTION WAITING FOR THE I/O PACKET THAT WAS JUST INPUT ?
                                ;
001164  032763  000200  000034          BIT     #FRFUC,MCR(R3)
001172  001534                          BEQ     90$                     ;NO, RETURN, RCV ISR WILL HANDLE
```

```
624 001174                                      CALL    SETAPR                  ;YES, MAP INTO TASKS AREA
625 001200  016320  000024                      MOV     R,EF0(R3),(R0)+         ;MOVE 36 BIT INPUT FUNCTION INTO..
626 001204  016320  000022                      MOV     R,EF1(R3),(R0)+         ;TASK'S BUFFER
627 001210  016310  000020                      MOV     R,EF2(R3),(R0)
628 001214  012667  000000G                     MOV     (SP)+,KISAR6            ;RESTORE THE OLD APR 6
629                                     ;
630                                     ;   SET UP RECEIVE PACKING MODE BITS (0,1 AND 2) IN MCR
631                                     ;
632 001220  010146                              MOV     R1,-(SP)                ;SAVE R1
633 001222  016300  000024                      MOV     R,EF0(R3),R0            ;R0 = BITS 32-47
634 001226  016301  000022                      MOV     R,EF1(R3),R1            ;R1 = BITS 16-31
635 001232  073027  000002                      ASHC    #2,R0                   ;R0 = TRANSACTION TYPE --
636                                                                             ;  -- BITS 30-35
637 001236  042700  177700                      BIC     #177700,R0              ;JUST TO MAKE SURE
638 001242  005067  176540                      CLR     NB,UHL                  ;NO UHL BLOCKS
639 001246  022700  000005                      CMP     #5,R0
640 001252  001003                              BNE     4$                      ;IF NOT QUERY, IGNORE UHL
641 001254  012767  177777  176524              MOV     #-1,NB,UHL              ;IF QUERY, SET UHL FLAG
642 001262  116000  000072'             4$:     MOVB    PACK(R0),R0             ;R0 = PACKING MODE
643 001266  042700  177400                      BIC     #177400,R0
644                                     ;
645 001272  042763  000007  000034              BIC     #7,MCR(R3)              ;CLEAR PREVIOUS PACKING MODE BITS
646 001300  050063  000034                      BIS     R0,MCR(R3)              ;SET PACKING MODE BITS
647 001304  012601                              MOV     (SP)+,R1                ;RESTORE R1
648                                     ;
649 001306  042763  000200  000034              BIC     #FRFUC,MCR(R3)          ;CLEAR THE INPUT NOTICE BIT
650                                     ;
651                                     ; GET THE DMA BLOCK COUNT FROM THE 36 BIT FUNCTION THAT WAS INPUT
652                                     ;
653 001314  016300  000022                      MOV     R,EF1(R3),R0
654 001320  042700  140000                      BIC     #140000,R0
655 001324  072027  177776                      ASH     #-2,R0
656 001330  010065  000000G                     MOV     R0,U.CW3(R5)
657 001334  005767  176446                      TST     NB,UHL
658 001340  001410                              BEQ     5$                      ;IGNORE IF UHL FLAG NOT SET
659 001342  010067  176440                      MOV     R0,NB,UHL               ;CALCULATE UHL BLOCK SIZE
660 001346  016300  000020                      MOV     R,EF2(R3),R0            ;  QUERY SIZE
661 001352  042700  177400                      BIC     #177400,R0
662 001356  160067  176424                      SUB     R0,NB,UHL
663                                     ;
664                                     ; CHECK TO SEE IF THE INPUT FUNCTION WAS A RESTART REQUEST
665                                     ;
666 001362                              5$:     SAVE    R1
667 001364  016300  000024                      MOV     R,EF0(R3),R0            ;R0 = BITS 32-35
668 001370  016301  000022                      MOV     R,EF1(R3),R1            ;R1 = BITS 16-31
669 001374  073027  000002                      ASHC    #2,R0                   ;R0=INPUT FUNCTION (BITS 30-35)
670 001400  022700  000052                      CMP     #RSTRTC,R0              ;RESTART REQUEST?
671 001404  001006                              BNE     3$                      ;NO IT WAS NOT
672 001406                                      RESTOR  R1
673 001410  012700  000000G                     MOV     #IE,CNR,R0              ;YES, RETURN AN ERROR
674 001414                                      CALL    IODONE
675 001420  000406                              BR      50$                     ;GO PROCESS THE REST OF THE ERROR
676 001422                              3$:     RESTOR  R1
677 001424  012700  000000G                     MOV     #IS,SUC,R0              ;GOOD INPUT, GIVE GOOD STATUS RETURN
678 001430                                      CALL    IODONE
679 001434  000415                              BR      99$                     ;GO RETURN TO CALLING ROUTINE
680                                     ;
```

```
                                        ; AN ERROR WAS DETECTED.  CAUSE AN ERROR INTERRUPT IN ON THE TRANSMIT
                                        ; SIDE OF THE HANDLER.
                                        ;
001436                                  50$:
001436  042763  000100  000010                BIC     #INT.T,CSR(R3)          ;DISABLE XMIT AND RCV INTS.
001444  042763  000100  000000                BIC     #INT.R,CSR(R3)
001452  012700  000002                        MOV     #XMT.R0                 ;TELL XMIT THAT ERROR WAS DETECTED
001456                                        CALL    SETERR
001462  000402                              BR      99$
                                        ;
                                        ; DO NOT ALLOW TIME OUTS FOR RECEPTION OF FUNCTION WORDS
                                        ;
001464                                  90$:
001464  105064  000000G                       CLRB    S.CTM(R4)               ;CLEAR TIME OUT COUNT.  THIS ...
                                                                                ;...INHIBITS THE TIME OUT...
                                                                                ;...MECHANIZM.
                                        ;
                                        ; RETURN TO CALLING ROUTINE
                                        ;
001470                                  99$:
001470                                        RETURN
```

```
                                        .SBTTL XMTINT ($ACOUT) --  ENTRY POINT
                                ;+
                                ;
                                ; **--XMTINT ($ACOUT)  --  ENTRY POINT
                                ;
                                ; THE PURPOSE OF THE "XMTINT" ROUTINE IS TO HANDLE THE INTERRUPTS
                                ; THAT RESULT FROM SENDING EITHER A 36 BIT STATUS WORD OR BLOCK
                                ; OF WORDS (16 BIT) VIA THE DMA.
                                ;
                                ; THE "XMTINT" ISR HANDLES TWO TYPES OF INTERRUPTS:
                                ;
                                ;       1) DMA DONE.
                                ;       2) STATUS SENT.
                                ;
                                ; INPUTS:
                                ;       AFTER THE CALL TO INTSV$ R4 CONTAINS THE CONTROLLER INDEX
                                ;       AND R5 CONTAINS A POINTER TO THE UCB.
                                ;
                                ; REGISTER USAGE:
                                ;       R4 AND R3 POINT TO THE FIRST ACC HARDWARE REGISTER.
                                ;       R2 CONTAINS THE ADDRESS OF EITHER THE FRIST XMIT OR RCV
                                ;        CSR.
                                ;       R0 CONTAINS AN ERROR CODE WHEN "IODONE" IS CALLED.
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
                                ;
001472                          $ACOUT::                        ;ENTRY POINT FOR THE EXECUTIVE
001472                          XMTINT:                         ;;;INTERNAL ACC HANDLER ENTRY POINT
001472                                  INTSV$  AC,PR5,A$$C11   ;;;GET UCB ADDRESS
001476  012704  000002                  MOV     #2,R4           ;;;FORCE R4 TO BE XMIT OFFSET IN CMTBL
001502  016405  000002'                 MOV     CNTBL(R4),R5    ;;;R5->XMIT UCB
001506  016504  000000G                 MOV     U.SCB(R5),R4    ;;;R4->SCB
001512  016404  000000G                 MOV     S.CSR(R4),R4    ;;;R4->FIRST ACC HARDWARE REGISTER
001516  042764  000100  000010         BIC     #INT.T,CSR(R4)  ;;;DISABLE BOTH ACC RCV AND XMIT...
001524  042764  000100  000000         BIC     #INT.R,CSR(R4)  ;;;...INTERRUPTS.
001532                                  CALL    $FORK           ;MAKE THE HANDLER INTERRUPTABLE.
001536  010403                          MOV     R4,R3           ;R3->FRIST ACC HARDWARE REGISTER
001540  010402                          MOV     R4,R2           ;R2->FIRST ACC HARDWARE XMIT
001542  062702  000010                 ADD     #T.CSR,R2       ;...CSR.
001546  005767  176314                 TST     RSTADD          ;RESTART?
001552  001046                          BNE     2$              ;YES
                                ;
                                ;   IGNORE SPURIOUS INTERRUPT
                                ;
001554  016504  000000G                 MOV     U.SCB(R5),R4
001560  105764  000000G                 TSTB    S.STS(R4)       ;IF UNIT IS NOT ACTIVE,
001564  001441                          BEQ     2$              ;  IGNORE INTERRUPT
                                ;
                                ; CHECK TO SEE IF THERE WAS AN ERROR
                                ;
001566                                  CALL    CKERR
0C1572  103005                          BCC     1$              ;NO, THERE WAS NOT AN ERROR
                                ;
                                ;AN ERROR WAS DETECTED, INDUCE AN ERROR INTO THE RCV SIDE
```

```
                                        ;
001574  012700  000000                          MOV     #RCV,R0                 ;TELL RCV THAT ERROR WAS DETECTED
001600                                          CALL    SETERR
001604  000437                                  BR      99$                     ;GO GET ANOTHER PACKET
                                        ;
                                        ;   NO I/O ERRORS
                                        ;
001606  016401  000000G                 1$:     MOV     S.PKT(R4),R1
001612  126127  000001G 000000C                 CMPB    I.FCN+1(R1),#IO.WLB/256,        ;WLB ?
001620  001013                                  BNE     3$                      ;BRANCH IF NO
001622  032763  000200  000010                  BIT     #DMADON,T.CSR(R3)       ;DMA COMPLETED?
001630                                          BOFF    2$                      ;BRANCH IF NO
001632  032763  001000  000010                  BIT     #WCEZ,T.CSR(R3)
001640                                          BOFF    2$
001642  005365  000000G                         DEC     U.CW3(R5)               ;DEC OUTPUT BLOCK COUNT
001646  000404                                  BR      4$                      ;THEN COMPLETE I/O
                                        ;
                                        ;   FUNCTION COMPLETION ?
                                        ;
001650  032763  100000  000034          3$:     BIT     #SSTU,MCR(R3)           ;FUNCTION SENT?
001656                                          BON     2$                      ;BRANCH IF NOT YET
001660  012700  000000G                 4$:     MOV     #IS.SUC,R0              ;RETURN GOOD STATUS
001664                                          CALL    IODONE
                                        ;
001670                                  2$:
001670  052763  00A100  000010                  BIS     #INT,T.CSR(R3)          ;ENABLE INTERRUPTS
001676  052763  000100  000000                  BIS     #INT.R,CSR(R3)
                                        ;
                                        ; GO AND SEE IF THERE ARE ANYMORE I/O PACKETS TO BE PROCESSED FOR THIS
                                        ; (POINTED TO BY R5) UNIT.
                                        ;
001704                                  99$:
001704  000167  176206                          JMP     ACINI
```

```
                                        .SBTTL  RCVINT ($ACINP) -- ENTRY POINT
                                ;+
                                ;
                                ; **--RCVINT ($ACINP)  --  ENTRY POINT
                                ;
                                ; THE PURPOSE OF THE "RCVINT" ROUTINE IS TO SORT OUT FOUR TYPES OF
                                ; RECEIVE INTERRUPTS:
                                ;
                                ;       1) DMA DONE,
                                ;       2) FUNCTION INPUT
                                ;       3) THE UNIVAC IS TRYING TO SEND A BLOCK OF DATA BUT
                                ;          NO QIO HAS BEEN ISSUED, AND
                                ;       4) THE XMTINT ISR HAS GENERATED AN ERROR INTERRUPT.
                                ;
                                ; INPUTS:
                                ;       AFTER THE CALL TO INTSV$ R5 CONTAINS THE ADDRESS OF THE UCB
                                ;
                                ; REGISTER USAGE:
                                ;       R0 IS USED TO PASS ERROR CODES TO "IODONE"
                                ;       R5->UCB
                                ;       R3->FIRST ACC HARDWARE REGIESTER
                                ;       R4->SCB
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
                                ;
001710                          $ACINP::                        ;;;ENTRY POINT FOR RSX-11M
001710                          RCVINT:
001710                                  INTSV$  AC,PR7,A$$C11   ;;;SAVE INTERRUPT REGIESTERS
                                ; NEED TO MANIPULATE FORK QUEUE AT PRIORITY 7
001714  005004                          CLR     R4              ;;;FORCE R4 TO BE RCV OFFSET IN CNTBL
001716  016405  000002'                  MOV     CNTBL(R4),R5    ;;;R5->RCV UCB
                                ; SEE IF WE'RE ALREADY IN FORK  QUEUE -- IF SO DON'T FORK AGAIN
001722  010046                          MOV     R0,-(SP)
001724  016504  000000G                 MOV     U.SCB(R5),R4    ;ADDRESS OF OUR
001730  062704  000000G                 ADD     #S.FRK,R4       ;  FORK PACKET
001734  016700  000000G                 MOV     $FRKHD,R0       ;START OF FORK QUEUE
001740  001407                          BEQ     ITSOK           ;BRANCH IF QUEUE EMPTY
001742  020004                  1$:     CMP     R0,R4           ;BRANCH IF NOT
001744  001003                          BNE     2$              ;  OUR ENTRY
001746  012600                          MOV     (SP)+,R0        ;WE'RE IN --
001750  000167  000000G                 JMP     $INTXT          ;  DON'T DO IT AGAIN
                                ;
001754  011000                  2$:     MOV     (R0),R0         ;CONTINUE TO END
001756  001371                          BNE     1$              ;  OF FORK QUEUE
001760  012600                  ITSOK:  MOV     (SP)+,R0        ;WE'RE NOT IN THE QUEUE
                                ;
001762  016504  000000G                 MOV     U.SCB(R5),R4    ;;;R4->SCB
001766  016404  000000G                 MOV     S.CSR(R4),R4    ;;;R4->FRIST ACC HARDWARE REGISTER
001772  042764  000100  000010          BIC     #INT.T,CSR(R4)  ;;;DISABLE XMIT AND RCV INTERRUPTS
002000  042764  000100  000000          BIC     #INT.R,CSR(R4)  ;;;
002006                                  CALL    $FORK           ;FORK TO AN INTERRUPTABLE PROCESS
002012  010403                          MOV     R4,R3           ;R3-> BASE ACC HARDWARE REGISTER
                                ;
                                ; CHECK TO SEE IF THERE WAS AN HARDWARE ERROR
```

```
                                        ;
002014  010302                                 MOV     R3,R2                   ;R2->FRIST ACC HARDWARE REGISTER
002016  062702  000000                          ADD     #R.CSR,R2               ;R2->RCV CSR
002022                                          CALL    CKERR                   ;CHECK FOR AN HARDWARE REGISTER
002026  103006                                  BCC     1$                      ;NO ERROR
                                        ;
                                        ; ERROR DETECTED.  INVOKE AN ERROR IN THE XMIT SIDE OF THE HANDLER IF
                                        ; THE RESTART FLAG IS NOT SET.
                                        ;
002030  012700  000002                          MOV     #XMT,R0                 ;TELL XMIT THAT ERROR WAS DETECTED
002034                                          CALL    SETERR
002040  000167  000530                          JMP     99$
                                        ;
                                        ; CHECK TO SEE IF A DMA OR FUNCTION WAS INPUT
                                        ;
002044                                  1$:
002044  032763  010000  000000                  BIT     #DPBF,R.CSR(R3)         ;FUNC OVERRODE THE DMA ?
002052                                          BON     8$                      ;BRANCH IF YES
002054  032763  000200  000034                  BIT     #FRFUC,MCR(R3)          ;FUNC INPUT ?
002062                                          BON     2$                      ;YES
                                        ;
                                        ; DMA WAS INPUT.  IF BIT 6 OF THE MCR IS SET IT INDICATES THAT THE
                                        ; UNIVAC IS TRYING TO SEND A DMA BLOCK BUT THERE IS NO ACTIVE QIO,
                                        ; IF THIS IS TRUE IT IS IGNORED UNTIL THE TASK ISSUES A QIO.
                                        ;
002064  032763  000100  000034          11$:    BIT     #DA,MCR(R3)             ;IS BIT SET?
002072  001413                                  BEQ     3$                      ;NO
002074  042763  000100  000034                  BIC     #DA,MCR(R3)             ;YES, NOW CLEAR THE BIT
002102  052763  000100  000010                  BIS     #INT.T,CSR(R3)          ;NOW ENABLE INTERRUPTS...
002110  052763  000100  000000                  BIS     #INT,R.CSR(R3)          ;...
002116  000167  000452                          JMP     99$
002122                                  3$:
002122                                          CALL    DMA
002126  000167  000442                          JMP     99$
                                        ;
                                        ; FUNCTION WAS INPUT.  IF IT WAS A RESTART REQUEST AND THE HANDLER IS NOT
                                        ; IN THE RESTART MODE, THEN SET THE RESTART FLAG AND NOTIFY THE XMIT SIDE
                                        ; VIA N INTERRUPT FROM BIT 13 OF THE XMIT CSR.  IF WE ARE ALREAD IN A
                                        ; RESTART MODE, RETURN A BAD STATUS AND WAIT FOR A TASK TO CAUSE A RESTART.
                                        ;
002132                                  2$:     SAVE    R0,R1
002136  016300  000024                          MOV     R.EF0(R3),R0            ;R0 = BITS 32-35
002142  016301  000022                          MOV     R.EF1(R3),R1            ;R1 = BITS 16-31
002146  073027  000002                          ASHC    #2,R0                   ;R0=INPUT FUNCTION (BITS 30-35)
002152  022700  000052                          CMP     #RSTRTC,R0              ;RESTART REQUEST ?
002156  001016                                  BNE     5$                      ;NO
002160                                          RESTOR  R0,R1
002164  012700  000000G                  8$:     MOV     #IE.CNR,R0              ;YES, RETURN ERROR STATUS
002170                                          CALL    IODONE
002174  005767  175664                          TST     RESTRT                  ;IS THE RESTART FLAG SET
002200  001167                                  BNE     31$                     ;YES, GET READY TO EXIT
002202  012700  000002                          MOV     #XMT,R0                 ;TELL XMIT THAT ERROR WAS DETECTED
002206                                          CALL    SETERR
002212  000562                                  BR      31$
                                        ;
                                        ; CHECK TO SEE IF WE HAVE A RESTART ACKNOWLEDGEMENT.  IF NOT IT IS A NORMAL
                                        ; FUNCTION INPUT.  THIS INTERRUPT ENTRY POINT CAN GET AN EXTRA RESTART ACK.
```

```
                                        ; THIS OCCURS WHEN BOTH THE UNIVAC AND PDP RESTART THE LINE AT THE SAME TIME.
                                        ; THE PDP WILL RETURN A BAD STATUS TO THE APPLICATIONS TASK, THE TASK WILL
                                        ; RE-INITIALIZE THE CHANNEL AND SEND ANOTHER RESTART REQUEST TO THE UNIVAC.
                                        ; IN THE MEAN TIME THE UNIVAC HAS DETECTED THE FIRST RESTART REQUEST SENT
                                        ; BY THE PDP.  SO IT SENDS A RESTART ACK.  THE PDP ACCEPTS THIS AS AN ACK
                                        ; TO ITS SECOND RESTART REQUEST.  THEN THE UNIVAC DETECTES THE SECOND RESTART
                                        ; REQUEST AND SENDS A SECOND RESTART ACK.  THIS ACK IS IGNORED SINCE THE
                                        ; PDP IS NO LONGER IN A RESTART MODE.
                                        ;
002214                                  5$:
002214  022700  000070                          CMP     #RACK,R0                ;IS THIS A RESTART ACK ?
002220  001536                                 BEQ     6$                      ;YES
002222                                          RESTOR  R0,R1
002226  016504  000000G                         MOV     U.SCB(R5),R4            ;NO, R4->SCB
002232  105764  000000G                         TSTB    S.STS(R4)               ;IS THE UNIT BUSY ?
002236  001520                                 BEQ     7$                      ;NO, WAIT UNTIL QIO ISSUED
002240  016401  000000G                         MOV     S.PKT(R4),R1
002244  126127  000001G 000007                 CMPB    I.FCN+1(R1),#IO.RFC/256,        ;FUNCTION ?
002252  001304                                 BNE     11$                     ;BRANCH IF NO
002254                                          CALL    SETAPR                  ;MAP INTO TASKS FUNCTION BUFFER
002260  016320  000024                         MOV     R.EF0(R3),(R0)+         ;MOVE 36 BIT BUFFER
002264  016320  000022                         MOV     R.EF1(R3),(R0)+
002270  016310  000020                         MOV     R.EF2(R3),(R0)
002274  012667  000000G                         MOV     (SP)+,KISAR6            ;RESTORE APR 6
002300  032763  010000  000000                 BIT     #DPBF,R.CSR(R3)         ;IF FUNCTION OVER WROTE DMA
002306  001411                                 BEQ     10$                     ;NO
002310  012700  000000G                         MOV     #IE.CNR,R0              ;YES, REPORT ERROR
002314                                          CALL    IODONE
002320  012700  000002                         MOV     #XMT,R0                 ;INDUCE ERROR IN XMIT SIDE
002324                                          CALL    SETERR
002330  000513                                 BR      31$
                                        ;
                                        ;   SET UP RECEIVE PACKING MODE BITS (0,1 AND 2) IN MCR
                                        ;
002332  010146                          10$:   MOV     R1,-(SP)                ;SAVE R1
002334  016300  000024                         MOV     R.EF0(R3),R0            ;R0 = BITS 32-47
002340  016301  000022                         MOV     R.EF1(R3),R1            ;R1 = BITS 16-31
002344  073027  0L.0002                        ASHC    #2,R0                   ;R0 = TRANSACTION TYPE
                                                                                ;  -- BITS 30-35
002350  042700  177700                         BIC     #177700,R0              ;JUST TO MAKE SURE
002354  005067  175426                         CLR     NB.UHL                  ;SHOW NO UHL
002360  022700  000005                         CMP     #5,R0                   ;QUERY ?
002364  001003                                 BNE     4$                      ;BRANCH IF NO
002366  012767  177777  175412                 MOV     #-1,NB.UHL              ;IF QUERY, SET UHL FLAG
002374  116000  000072'              4$:       MOVB    PACK(R0),R0             ;R0 = PACKING MODE
002400  042700  177400                         BIC     #177400,R0
                                        ;
002404  042763  000007  000034                 BIC     #7,MCR(R3)              ;CLEAR PREVIOUS PACKING MODE BITS
002412  050063  000034                         BIS     R0,MCR(R3)              ;SET PACKING MODE BITS
002416  012601                                 MOV     (SP)+,R1                ;RESTORE R1
                                        ;
002420  012700  000000G                         MOV     #IS.SUC,R0              ;RETURN GOOD STATUS
002424                                          CALL    IODONE
                                        ;
                                        ; GET THE INPUT BLOCK COUNT
                                        ;
002430  016300  000022                          MOV     R.EF1(R3),R0            ;GET THE WORD THAT CONTAINS THE FIELD
```

```
 965 002434  042700  140000                  BIC     #140000,R0              ;MASK OFF GARBAGE
 966 002440  072027  177776                  ASH     #-2,R0                  ;RIGHT JUSTIFY
 967 002444  010065  000000G                 MOV     R0,U.CW3(R5)            ;SAVE IT IN THE BLOCK COUNT
 968 002450  005767  175332                  TST     NB.UHL                  ;UHL FLAG SET ?
 969 002454  001433                          BEQ     30$                     ;BRANCH IF NO
 970 002456  010067  175324                  MOV     R0,NB.UHL               ;NB.UHL = TOTAL NO. OF BLOCKS
 971 002462  016300  000020                  MOV     R.EF2(R3),R0            ;R0 = NO. OF BLOCKS OOF QUERY TEXT
 972 002466  042700  177400                  BIC     #177400,R0              ;JUST TO MAKE SURE
 973 002472  160067  175310                  SUB     R0,NB.UHL               ;NO. OF UHL BLOCKS
 974 002476  000422                          BR      30$
 975                                 ;
 976                                 ; ENABLE INTERRUPTS
 977                                 ;
 978 002500                          7$:
 979 002500  052763  000100  000000          BIS     #INT,R.CSR(R3)
 980 002506  052763  000100  000010          BIS     #INT,T.CSR(R3)
 981 002514  000427                          BR      99$
 982                                 ;
 983                                 ; PROCESS A RESTART ACKNOWLEDGEMENT
 984                                 ;
 985 002516                          6$:     RESTOR  R0,R1
 986 002522  005767  175336                  TST     RESTRT                  ;IS THE RESTART FLAG SET ?
 987 002526  001406                          BEQ     30$                     ;NO, IGNORE THE ACK
 988 002530  012700  000000G                 MOV     #IS.SUC,R0              ;YES, RETURN GOOD STATUS
 989 002534                                  CALL    IODONE
 990 002540  005067  175320                  CLR     RESTRT                  ;SHOW WE ARE NOT IN RESTART MODE
 991                                 ;
 992                                 ;
 993 002544                          30$:
 994 002544  052763  000100  000010          BIS     #INT,T.CSR(R3)          ;ENABLE XMIT INTERRUPT
 995 002552  052763  000100  000000          BIS     #INT,R.CSR(R3)          ;ENABLE RCV INTERRUPT
 996 002560                          31$:
 997 002560  042763  000200  000034          BIC     #FRFUC,MCR(R3)          ;CLEAR FUNCTION INPUT BITS
 998 002566  042763  010000  000000          BIC     #DPBF,R.CSR(R3)
 999                                 ;
1000                                 ; RETURN
1001                                 ;
1002 002574                          99$:
1003 002574  000167  175316                  JMP     ACINI
```

```
                                        .SBTTL DMA -- SUBROUTINE
                                ;+
                                ;
                                ; **--DMA SUBROUTINE
                                ;
                                ; THE PURPOSE OF THE DMA ROUTINE IS TO PROCESS DMA RECEIVE INTERRUPTS.
                                ;
                                ; INPUTS:
                                ;       R3->FIRST ACC HARDWARE REGISTER
                                ;       R4->FIRST ACC HARDWARE REGISTER
                                ;       R5->UCB
                                ;
                                ; REGISTER USAGE:
                                ;       R0 CONTAINS ERROR CODES.
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
                                ;
002600                          DMA:
002600  016504  000000G         MOV     U.SCB(R5),R4
002604  105764  000000G         TSTB    S.STS(R4)               ;IF THE UNIT IS NOT ACTIVE,
002610  001442                  BEQ     99$                     ;  IGNORE SPURIOUS INTERRUPT.
002612  016401  000000G         MOV     S.PKT(R4),R1
002616  126127  000001G 000000C CMPB    I.FCN+1(R1),#IO.RLB/256.        ;RLB?
002624  001034                  BNE     99$                     ;IGNORE IF NOT RLB
002626  032763  000200  000000  BIT     #DMADON,R.CSR(R3)       ;DMA DONE INDICATION?
002634                          BOFF    99$                     ;NO, GO EXIT
002636  032763  001000  000000  BIT     #WCEZ,R.CSR(R3)         ;DMA WORD COUNT = 0?
002644                          BOFF    99$                     ;NO, GO EXIT
002646  012700  000000G         MOV     #IS.SUC,R0              ;YES, RETURN GOOD I/O STATUS
002652                          CALL    IODONE
002656  042763  001200  000000  BIC     #<DMADON+WCEZ>,R.CSR(R3);RESET DMA DONE BITS
002664  005365  000000G         DEC     U.CW3(R5)               ;DEC INPUT BLOCK COUNT
002670  001412                  BEQ     99$                     ;IF NO MORE BLOCKS
002672  026567  000000G 175106  CMP     U.CW3(R5),NB.UHL        ;SWITCH FROM QUERY TO UHL BLOCKS?
002700  001006                  BNE     99$                     ;NO
002702  042763  000007  000034  BIC     #7,MCR(R3)              ;YES: CHANGE PACKING MODE
002710  052763  000004  000034  BIS     #4,MCR(R3)
002716                          39$:
002716  052763  000100  000010  BIS     #INT,T.CSR(R3)          ;ENABLE INTERRUPTS
002724  052763  000100  000000  BIS     #INT,R.CSR(R3)
002732                          RETURN
```

```
                                        .SBTTL  POWER FAIL -- ENTRY POINT
                                ;+
                                ;
                                ; THE PURPOSE OF THIS ENTRY POINT IS TO RESET THE INTERFACE AND LOAD
                                ; CNTBL WITH VALID UCB ADDRESSES.
                                ;
                                ; INPUTS:
                                ;       R5->UCB
                                ;       R4->SCB
                                ;       R3=CONTROLLER INDEX
                                ;
                                ; REGISTER USAGE:
                                ;       R0->FIRST ACC HARDWARE REGISTER
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
                                ;
002734                          POWER:
002734  016400  000000G         MOV     S.CSR(R4),R0            ;R0->FIRST ACC HARDWARE REGISTER
002740  052760  000004  000010  BIS     #RESET,T.CSR(R0)        ;RESET THE INTERFACE
002746  005065  000000G         CLR     U.CW3(R5)               ;RESET THE DMA BLOCK COUNT
002752  010563  000002'          MOV     R5,CNTBL(R3)            ;PUT UCB ADDR INTO CNTBL
002756                          RETURN
```

```
                                        .SBTTL  TIME OUT -- ENTRY POINT
                                ;+
                                ;
                                ; THE TIME OUT ENTRY POINT IS CALLED AFTER AN EXCESSIVE AMOUNT OF TIME HAS
                                ; EXPIRED IN A DMA OR SEND STATUS OPERATION.  ONLY RECEIVE DMA OPERATIONS
                                ; ARE TIMED.  NOTHING IS DONE FOR THE RECEPTION OF FUCNTION WORDS BECAUSE
                                ; ONE CANNOT PREDICT  WHEN TRANSACTIONS ARE TO BE INITIATED BY THE HOST.
                                ;
                                ; INPUT:
                                ;       R0 = I/O STATUS CODE IE,DNR
                                ;       R2=ADDRESS OF CSR
                                ;       R3=CONTROLLER INDEX
                                ;       R4=ADDRESS OF THE SCB
                                ;       R5=ADDRESS OF THE UCB
                                ;
                                ;
                                ; REGISTER USAGE
                                ;       R0 CONTAINS ERROR CODES
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
                                ;
1100 002760                     TIMOUT:
1101 002760  005767  175104             TST     LOOPB                   ;IF IN LOOP BACK MODE DO NOT...
1102 002764  001006                     BNE     3$                      ;...CLEAR THE INTERRUPTS
1103 002766  042762  000100  000010     BIC     #INT.T,CSR(R2)          ;DISABLE TIME OUTS
1104 002774  042762  000100  000000     BIC     #INT.R,CSR(R2)
1105 003002                     3$:
1106 003002  005065  000000G            CLR     U.CW3(R5)               ;RESET THE DMA BLOCK COUNT
1107 003006  105765  000000G            TSTB    U.UNIT(R5)              ;WHICH UNIT ?
1108 003012  001405                     BEQ     1$                      ;RECEIVE
1109 003014  012700  000000             MOV     #RCV,R0                 ;TELL RCV THAT ERROR WAS FOUND
1110 003020                             CALL    SETERR
1111 003024  000404                     BR      2$
1112 003026                     1$:
1113 003026  012700  000002             MOV     #XMT,R0                 ;TELL XMIT THAT ERROR WAS FOUND
1114 003032                             CALL    SETERR
1115 003036                     2$:
1116 003036  012700  000000G            MOV     #IE.TMO,R0              ;RETURN ERROR CODE TO TASK
1117 003042                             CALL    IODONE
1118 003046  000167  175044             JMP     ACINI                   ;GO GET ANTHER PACKET
```

```
                                        .SBTTL CANCEL -- ENTRY POINT
                                ;+
                                ;
                                ; THE PURPOSE OF THE CANCEL ENTRY POINT IS TO KILL ON GOING I/O'S,
                                ;  THE CURRENT I/O IS KILLED AND AN ERROR INTERRUPT IS SET TO CAUSE
                                ; THE OTHER SIDE OF THE HANDLER TO ABORT ANY ACTIVE I/O IT MAY HAVE.
                                ;
                                ; INPUTS:
                                ;       R5->UCB
                                ;       R4->SCB
                                ;       R3=CONTROLLER INDEX
                                ;       R1->TCB OF CURRENT TASK
                                ;       R0->ACTIVE I/O PACKET
                                ;
                                ; REGISTER USAGE:
                                ;       R3->FIRST ACC HARDWARE REGISTER
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
                                ;
003052                          CANCEL:
003052  016403  000000G         MOV     S.CSR(R4),R3            ;R3->FIRST ACC HARDWARE REGISTER
003056  105064  000000G         CLRB    S.CTM(R4)               ;CANCEL ANY TIME OUT
003062  042763  000100  000010  BIC     #INT.T,CSR(R3)          ;DISABLE ACC INTERRUPTS
003070  042763  000100  000000  BIC     #INT.R,CSR(R3)
003076  012700  000000G         MOV     #IE.ABO,R0              ;RETURN ABORT STATUS TO TASK
003102                          CALL    IODONE
003106  005065  000000G         CLR     U.CW3(R5)               ;RESET THE DMA BLOCK COUNT
003112  105765  000000G         TSTB    U.UNIT(R5)              ;IS THE A RCV OR XMIT UNIT
003116  001005                  BNE     1$                      ;A XMIT UNIT
003120  012700  000002          MOV     #XMT,R0                 ;TELL XMIT THAT AN ERROR WAS FOUND
003124                          CALL    SETERR
003130  000404                  BR      99$
003132                          1$:
003132  012700  000000          MOV     #RCV,R0                 ;TELL RCV THAT AN ERROR WAS FOUND
003136                          CALL    SETERR
003142                          99$:
003142  000167  174750          JMP     ACINI                   ;GO SEE IF THERE ARE ANY MORE PACKETS
```

```
                        .SBTTL  CKERR -- ROUTINE
                ;+
                ;
                ; THE PURPOSE OF THE CKERR ROUTINE IS TO CHECK ERROR BITS IN A GIVEN
                ; CSR.
                ;
                ; INPUTS:
                ;       R5->UCB
                ;       R2->CSR
                ;
                ; REGISTER USAGE:
                ;       R0 PASSES ERROR CODES TO "IODONE"
                ;
                ; OUTPUTS:
                ;       NONE
                ;
                ;-
                ;
003146          CKERR:
003146  032712  004000          BIT     #PARITY,(R2)            ;IS THERE A PARITY ERROR ?
003152  001410                  BEQ     1$                      ;NO
003154  012760  000000G         MOV     #IE.VER,R0              ;YES, RETURN ERROR STATUS
003160                          CALL    IODONE
003164  005065  000000G         CLR     U.CW3(R5)               ;CLEAR THE DMA BLOCK COUNT
003170  000261                  SEC                             ;SET ERROR RETURN
003172  000426                  BR      99$
003174                  1$:
003174  032712  060000          BIT     #<NEMR+UNVCR>,(R2)      ;COMPOSIT ERROR BIT SET ?
003200  001002                  BNE     2$                      ;YES
003202  000241                  CLC
003204  000421                  BR      99$                     ;NO, GIVE NO ERROR RETURN
003206                  2$:
003206  032712  040000          BIT     #NEMR,(R2)              ;MEMORY ADDRESSING ERROR ?
003212  001405                  BEQ     3$                      ;NO
003214  012700  000000G         MOV     #IE.SPC,R0              ;YES, RETURN ERROR
003220                          CALL    IODONE
003224  000404                  BR      50$
003226                  3$:
003226  012700  000000G         MOV     #IE.CNR,R0
003232                          CALL    IODONE
003236                  50$:
003236  042712  060000          BIC     #<NEMR+UNVCR>,(R2)      ;CLEAR THE ERROR BITS
003242  005065  000000G         CLR     U.CW3(R5)               ;RESET THE DMA BLOCK COUNT
003246  000261                  SEC                             ;SET ERROR RETURN
003250                  99$:
003250                          RETURN
```

```
                                        .SBTTL  SETAPR -- ROUTINE
                                ;+
                                ;
                                ; **--SETAPR ROUTINE
                                ;
                                ; THE PURPOSE OF THE SETAPR ROUTINE IS TO CONVERT A PHYSICAL ADDRESS
                                ; TO A VIRTUAL ADDRESS.
                                ;
                                ; INPUTS:
                                ;       R5->UCB
                                ;
                                ; REGISTER USAGE:
                                ;       R0 IS USED FOR WORKING STORAGE AND IS DESTROYED
                                ;       R1 IS USED FOR WORKING STORAGE AND IS RESTORED
                                ;
                                ; OUTPUTS:
                                ;       R0 CONTAINS THE VIRTUAL ADDRESS OF THE USER BUFFER
                                ;       APR 6 IS REMAPPED TO THE TASK'S AREA
                                ;       (SP) CONTAINS THE OLD APR 6 VALUE
                                ;
                                ;-
                                ;
1230 003252                     SETAPR:
1231 003252  011646                     MOV     (SP),-(SP)              ;MAKE ROOM FOR THE OLD APR 6
1232 003254  010146                     MOV     R1,-(SP)                ;SAVE R1
1233 003256  016500  000000G            MOV     U.BUF(R5),R0            ;MOV UPPER 2 BITS OF 18 BIT ADDR
1234 003262  042700  177717             BIC     #177717,R0              ;GET RID OF GARBAGE
1235 003266  072027  177774             ASH     #-4,R0                  ;RIGHT JUSTIFY BITS 18 AND 17
1236 003272  016501  000002G            MOV     U.BUF+2(R5),R1          ;GET LOWER ORDER 16 BITS OF ADDR
1237 003276  073027  177772             ASHC    #-6,R0                  ;CONVERT TO 32 WORD APR ADDRESS
1238 003302  016766  000000G 000004     MOV     KISAR6,4(SP)            ;SAVE THE OLD APR 6
1239 003310  010167  000000G            MOV     R1,KISAR6               ;PUT THE NEW VALUE INTO APR 6
1240 003314  016500  000002G            MOV     U.BUF+2(R5),R0          ;GET LOWER ORDER 16 BITS
1241 003320  042700  177700             BIC     #177700,R0              ;MASK OFF UPPER 10 BITS
1242 003324  052700  140000             BIS     #140000,R0              ;SETUP FOR APR 6
1243 003330  012601                     MOV     (SP)+,R1                ;RESTORE R1
1244 003332                             RETURN
```

```
                                        .SBTTL  IODONE -- ROUTINE
                                ;+
                                ;
                                ; **--IODONE ROUTINE
                                ;
                                ; THE PURPOSE OF THIS ROUTIEN IS TO SEND AN I/O DONE STATUS BACK TO THE
                                ; APPLICATION TASK VIA THE EXECUTIVE.   THIS IS DONE ONLY IF THERE IS AN
                                ; ACTIVE I/O FOR THE DESIRED UNIT.
                                ;
                                ; INPUTS:
                                ;       R5->UCB
                                ;       R0=ERROR CODE
                                ;
                                ; REGISTER USEAGE:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
                                ;
003334                          IODONE:
003334  010346                         MOV     R3,-(SP)                ;SAVE R3 - R5
003336  010446                         MOV     R4,-(SP)
003340  010546                         MOV     R5,-(SP)
003342  005767  174520                 TST     RSTADD                  ;IS THE RESTART ADDRESS ZERO ?
003346  001407                         BEQ     1$                      ;YES
003350  016705  174512                 MOV     RSTADD,R5               ;R5->RESTART UCB
003354                                  CALL    $IODON                  ;SEND STATUS
003360  005067  174502                 CLR     RSTADD                  ;CLEAR THE RESTART ADDRESS
003364  000407                         BR      99$
003366                          1$:
003366  016504  000000G                MOV     U.SCB(R5),R4
003372  105764  000000G                TSTB    S.STS(R4)               ;IS THE UNIT ACTIVE ?
003376  001402                         BEQ     99$                     ;NO
003400                                  CALL    $IODON                  ;YES, RETURN STATUS
003404                          99$:
003404  012605                         MOV     (SP)+,R5                ;RESTORE R3 - R5
003406  012604                         MOV     (SP)+,R4
003410  012603                         MOV     (SP)+,R3
003412                                  RETURN
```

```
                                        .SBTTL  SETERR --- ROUTINE
                                ;+
                                ;
                                ; **--SETERR ROUTINE
                                ;
                                ; THE PURPOSE OF THIS ROUTINE IS TO  SET A TIME OUT IN THE OTHER UNITS SCB.
                                ; THIS IS DONE IF AN ERROR IS ENCOUNTERED IN THE CURRENT UNIT.  THE ERROR
                                ; TIME OUT IS SET ONLY IF THE OTHER UNIT HAS AN ACTIVE I/O
                                ;
                                ; INPUTS:
                                ;       R0 CONTAINS THE OFFSET INTO "CNTBL" FOR THE CORRECT UCB
                                ;
                                ; REGISTER USAGE:
                                ;       R4 AND R5 ARE SAVE AND RESTORED.  R4->SCB, AND R5->UCB
                                ;       OF THE OTHER UNIT.
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
                                ;
003414                          SETERR:
003414  010546                          MOV     R5,-(SP)                ;SAVE R5 AND R4
003416  010446                          MOV     R4,-(SP)
003420  005267  174440                  INC     RESTRT                  ;SET THE RESTART FLAG
003424  016005  000002'                 MOV     CNTBL(R0),R5            ;R5->OTHER UNITS UCB
003430  016504  000000G                 MOV     U.SCB(R5),R4            ;R4->OTHER UNITS SCB
003434  105764  000000G                 TSTB    S.STS(R4)               ;IS THE OTHER UNIT BUSY ?
003440  001405                          BEQ     99$                     ;NO, DO NOTHING
003442  112764  000001  000000G        MOVB    #1,S.CTM(R4)            ;SET TIME OUT TO OCCUR WITHIN
                                                                        ;ONE SECOND
003450  005065  000000G                 CLR     U.CW3(R5)               ;RESET THE DMA BLOCK COUNT
003454                          99$:
003454  012604                          MOV     (SP)+,R4                ;RESTORE R4 AND R5
003456  012605                          MOV     (SP)+,R5
003460                                  RETURN
                                ;***--***
000001                                  .END
```

ACC HANDLER MACRO M1110 27-MAR-1978 [illegible]
SYMBOL TABLE

```
ACINI   000116R
A$$CHK= 000000
A$$CPS= 000000
A$$C11= 000002 G
A$$NSI= 000000
A$$PRI= 000000
A$$TRP= 000000
BBPDP = 000004
BBUNVC= 000010
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BYTE  = 000002
BYTE0 = 000000
BYTE1 = 000001
BYTE2 = 000002
BYTE3 = 000003
BYTE4 = 000004
BYTE5 = 000005
BYTE6 = 000006
BYTE7 = 000007
BYTE8 = 000010
BYTE9 = 000011
BYTVAL= 000012
CANCEL  003052R
CKERR   003146R
CNTBL   000002R
COMP  = 100000
C$$CDA= 000002
C$$CKP= 000000
C$$CSR= 177404
C$$INT= 000000
C$$ORE= 002000
C$$RSH= 177564
C$$RUN= 000000
C$$TTY= 177564
DA    = 000100
DBSLEN= 000116
DMA     002600R
DMADON= 000200
DMAGO = 000001
DPBF  = 010000
D$$ISK= 000000
D$$L11= 000004
D$$SHF= 000000
D$$YNC= 000000
D$$YNM= 000000
EA16  = 000020
EA17  = 000040
E$$XPR= 000000
FD.FID  000000      003
FD.FNB  000006      003
FD.FVR  000004      003
FD.LEN  000010      003
FRFUC = 000200
FUNCT   001130R
F$$LVL= 000001
G$$TPP= 000000
G$$TSS= 000000
G$$TTK= 000000
G$$WRD= 000000
H$$RTZ= 000074
IE.ABO= ****** GX
IE.BAD= ****** GX
IE.CNR= ****** GX
IE.DNR= ****** GX
IE.IFC= ****** GX
IE.ILU= ****** GX
IE.LNL= ****** GX
IE.SPC= ****** GX
IE.TMO= ****** GX
IE.VER= ****** GX
INT   = 000100
IODONE  003334R
IO.ADS= 014000
IO.ATA= 001410
IO.CAS= 015420
IO.CBO= 015510
IO.CCI= 014000
IO.CCO= 000440
IO.CIN= 016500
IO.CLK= 015000
IO.CNT= 017000
IO.CON= 015400
IO.CPR= 015410
IO.CPW= 016520
IO.CRC= 001040
IO.CRJ= 015440
IO.CSI= 013000
IO.CSM= 016470
IO.CTI= 015400
IO.CTL= 016400
IO.CTR= 015610
IO.CTY= 003400
IO.DCI= 014400
IO.DIS= 016000
IO.DLB= 016540
IO.DSI= 013400
IO.DTI= 016000
IO.DTY= 006400
IO.EOF= 003000
IO.FDX= 003020
IO.FLN= 012400
IO.GLC= 016430
IO.GLI= 016420
IO.GNI= 016410
IO.GRC= 016450
IO.GRI= 016440
IO.GRN= 016460
IO.GTS= 002400
IO.HDX= 003010
IO.HIS= 015000
IO.HNG= 003000
IO.INI= 014400
IO.INL= 002400
IO.ITI= 017000
IO.LDI= 007000
IO.LED= 012000
IO.LEI= 007410
IO.LKE= 012000
IO.LOD= 014000
IO.LSI= 011000
IO.LTI= 007400
IO.LTY= 010000
IO.MAO= 003410
IO.MCS= 013400
IO.MDA= 016000
IO.MDI= 014400
IO.MDO= 015400
IO.MLO= 006000
IO.MOD= 003000
IO.MSO= 005000
IO.NLB= 016530
IO.NLK= 011400
IO.ONL= 017400
IO.RAD= 010400
IO.RAL= 001010
IO.RBC= 003000
IO.RCI= 015000
IO.RCV= 015000
IO.RDB= 001200
IO.RDD= 010010
IO.REL= 013400
IO.RFC= 003400
IO.RHD= 001010
IO.RLB= ****** GX
IO.RLV= 001100
IO.RMT= 010020
IO.RNC= 001040
IO.RNE= 001020
IO.RNS= 001020
IO.RPB= 001040
IO.RPR= 004400
IO.RST= 001001
IO.RTC= 003400
IO.RTI= 016400
IO.RTM= 001200
IO.RWD= 002400
IO.RWU= 002540
IO.RIC= 002400
IO.SAO= 004000
IO.SCS= 013000
IO.SDI= 013000
IO.SDO= 012400
IO.SEC= 002520
IO.SLO= 005400
IO.SMO= 002560
IO.SPB= 002420
IO.SPF= 002440
IO.SPW= 016510
IO.SSO= 004400
IO.STA= 015400
IO.STC= 002500
IO.STP= 016400
IO.SYN= 003040
IO.TFC= 003000
IO.TRM= 002410
IO.UDI= 011410
IO.UEI= 011450
IO.UER= 011440
IO.USI= 011460
IO.UTI= 011420
IO.UTY= 011430
IO.WAL= 000410
IO.WBT= 000500
IO.WDD= 000444
IO.WLB= ****** GX
IO.WLC= 000420
IO.WLS= 000410
IO.WLT= 000410
IO.WLV= 000500
IO.WMS= 000420
IO.WNS= 000420
IO.WPB= 000440
IO.XMT= 014400
IO.XNA= 014410
IS.SUC= ****** GX
ITSOK   001760R
I$$RAR= 000000
I$$RDN= 000000
I.FCN = ****** GX
I.PRM = ****** GX
KISAR6= ****** GX
K$$CNT= 177546
K$$CSR= 177546
K$$IEN= 000115
K$$LDC= 000001
K$$TPS= 000074
LD$AC = 000001
LD$XM = 000000
LOOPB   000070R
L$$ASG= 000000
L$$DRV= 000000
L$$PTO= 000144
L$$P11= 000001
MC      000036      004
MCR     000034      004
M$$CRB= 000124
M$$CRX= 000000
M$$FCS= 000000
M$$MGE= 000000
M$$MUP= 000000
M$$OVR= 000000
N     = 000012
NB.UHL  000006R
NEMR  = 040000
N$$LDV= 000001
N$$MOV= 000041
N.BFAC= 000004
N.BHGH= 000006
N.BTCH= 000004
N.BUFB= 004000
N.BUFW= 002000
N.FOS = 000764
N.QURY= 000031
N.SUNT= 000002
PACK    000072R
PARITY= 004000
POWER   002734R
P$$GMX= 000000
P$$LAS= 000000
P$$P45= 000000
P$$RFL= 000000
P$$RTY= 000000
P$$SRF= 000000
P$$URD= 000000
Q$$OPT= 000007
RACK  = 000070
RBF   = 000400
RCV   = 000000
RCVINT  001710R
RECEVE  000740R
RESET = 000004
RESTRT  000064R
RETTE   000252R
RPM0  = 000001
RPM1  = 000002
RPM2  = 000004
RSTART  000066R
RSTRT   000254R
R$TRTC= 000052
R$$DER= 000000
R$$EXV= 000000
R$$K11= 000001
R$$P11= 000001
R$$SND= 000000
R$$11M= 000000
R.CSR   000000      004
R.DBUF  000002      004
R.EFLG  000024      004
R.EF1   000022      004
```

```
R.EF2   000020      004 SR.NIP  000022      002 TPM1   = 001000      T$$30P= 000000          WORD4  = 000010
R.MA    000004      004 SR.SDB  000032      002 TPM2   = 002000      T.CSR   000010      004 WORD5  = 000012
R.WC    000006      004 SR.SRC  000002      002 T$$ACR= 000000      T.DB    000012      004 WORD6  = 000014
SETAPR  003252R         SR.SUN  000000      002 T$$BTW= 000000      T.EF0   000032      004 WORD7  = 000016
SETERR  003414R         SR.TWS  000056      002 T$$BUF= 000000      T.EF1   000030      004 WORD8  = 000020
SR.ARE  000114      002 SR.WSL  000052      002 T$$CCA= 000000      T.EF2   000026      004 WORD9  = 000022
SR.ARS  000106      002 SR.YR   000004      002 T$$CCO= 000000      T.MA    000014      004 WRDVAL= 000024
SR.DAY  000010      002 SR.1IN  000024      002 T$$CTR= 000000      T.WC    000016      004 XMIT    000464R
SR.DLT  000014      002 SR.1IP  000016      002 T$$GMC= 000000      UNVCR = 020000          XMT    = 000002
SR.ECB  000047      002 SSTU    = 100000        T$$GTS= 000000      U.BUF = ****** GX       XMTINT  001472R
SR.ECH  000046      002 S$$YSZ= 007600          T$$KMG= 000000      U.CW3 = ****** GX       X$$DBT= 000000
SR.ECL  000050      002 S.CSR = ****** GX       T$$LWC= 000000      U.SCB = ****** GX       X$$M11= 000002
SR.FIB  000012      002 S.CTM = ****** GX       T$$M11= 000001      U.UNIT= ****** GX       $ACINP  001710RG
SR.GRE  000100      002 S.FRK = ****** GX       T$$RNE= 000000      VEC     000010R         $ACOUT  001472RG
SR.GRS  000072      002 S.ITM = ****** GX       T$$RPR= 000000      V$$CTR= 000404          $ACTBL  000054RG
SR.LEN  000122      002 S.PKT = ****** GX       T$$RST= 000000      V$$RSN= 000031          $FORK  = ****** GX
SR.LIN  000066      002 S.TS  = ****** GX       T$$RUB= 000000      WCEZ  = 001000          $FRKHD= ****** GX
SR.LIP  000062      002 TBE     = 000400        T$$SYN= 000000      WORD0 = 000000          $GTPKT= ****** GX
SR.MON  000006      002 TEMP    000000R         T$$TRW= 000000      WORD1 = 000002          $INTXT= ****** GX
SR.NDC  000042      002 TIMOUT  002760R         T$$UTB= 000000      WORD2 = 000004          $IODON= ****** GX
SR.NDS  000036      002 TPM0    = 000400        T$$VBF= 000000      WORD3 = 000006          ...GBL= 000000
SR.NIN  000030      002

. ABS.  000000    000
        003462    001
SRCOFF  000122    002
FDSCOF  000010    003
ACCREG  000040    004
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  10656 WORDS  (42 PAGES)
DYNAMIC MEMORY:  12308 WORDS  (47 PAGES)
ELAPSED TIME:  00:01:01
ACDRV,ACDRV/-SP=[1,1]EXEMC/ML,[200,200]RSXMC/PA:1,[20,1]P,ACDRV
```

```
                                        .TITLE  ACC HANDLER DATA BASE
                                        .MCALL  DEVDF$,HWDDF$
000000                                  DEVDF$
000000                                  HWDDF$
        000002                  A$$C11  =       2               ;NO OF UNITS
                                ;
                                ; START OF ACC DATA BASE
                                ;
000000                          $ACDAT::
                                ;
                                ; DEVICE CONTROL BLOCK (DCB)
                                ;
000000                          ACDCB:
000000  000000                          .WORD   0               ;D.LNK, LINK TO NEXT DCB
000002  000042'                         .WORD   .AC0            ;D.UCB, LINK TO FIRST UCB
000004     101     103                  .ASCII  /AC/            ;D.NAM, HANDLER NAME
000006     000     001                  .BYTE   0,1             ;D.UNIT, LOW AND HIGH UNIT NUMBERS
000010  000036                          .WORD   ACEND-ACSTRT    ;D.UCBL, UCB LENGTH
000012  000000                          .WORD   0               ;D.DSP, ADDRESS OF HANDLER DISPATCH TABLE
000014  000347                          .WORD   347             ;D.MSK, LEGAL FUNCTIONS:
                                                                ;       1) IO.KIL
                                                                ;       2) IO.WLB
                                                                ;       3) IO.RLB
                                                                ;       4) IO.INL
                                                                ;       5) IO.TFC
                                                                ;       6) IO.RFC
000016  000040                          .WORD   40              ;D.MSK, CONTROL FUNCTIONS
000020  000000                          .WORD   0               ;D.MSG, NO-OP'D MASK
000022  000000                          .WORD   0               ;D.MSK, ACP MASK
000024  000000  000000  000000          .WORD   0,0,0,0 ;D.MSK, FOR THE ABOVE WORDS ONLY, BITS 16-32
000032  000000
000034  000000                          .WORD   0               ;D.PCB, PARTITION CONTROL BLOCK
                                ;
                                ; UNIT CONTROL BLOCK (UCB) -----  UNIT 0
                                ;
000036                          ACSTRT:
000036  000000                          .WORD   0               ;U.LUIC
000040  000000                          .WORD   0               ;U.OWN
000042                          .AC0::
000042  000000'                          .WORD   ACDCB   ;U.DCB, ADDRESS OF DCB
000044  000042'                          .WORD   .-2     ;U.RED, REDIRECT POINTER
000046     321                          .BYTE   UC.PWF+UC.NPR+UC.ALG+1  ;U.CTL,
000047     000                          .BYTE   0       ;U.STS, UNIT STATUS
000050     000                          .BYTE   0       ;U.UNIT, UNIT NUMBER
000051     000                          .BYTE   0       ;U.ST2, UNIT STATUS
000052  000000                          .WORD   0       ;U.CW1
000054  000000                          .WORD   0       ;U.CW2
000056  000000                          .WORD   0       ;U.CW3, BLOCK COUNT FOR MULTI-BLOCK DPRT IO
000060  000000                          .WORD   0       ;U.CW4
000062  000132'                          .WORD   ACSCB0  ;U.SCB, POINTER TO UNIT'S SCB
000064  000000                          .WORD   0       ;U.ATT
000066  000000                          .WORD   0       ;U.BUF
000070  000000                          .WORD   0       ;U.CNT
000072  000000                          .WORD   0       ;
000074                          ACEND:
                                ;
                                ; UNIT CONTROL BLOCK (UCB) -----  UNIT 1
```

```
                                ;
000074  000000                          .WORD   0       ;U.LUIC
000076  000000                          .WORD   0       ;U.OWN
000100                          .AC1::
000100  000000'                         .WORD   ACDCB   ;U.DCB, ADDRESS OF DCB
000102  000100'                         .WORD   .-2     ;U.RED, REDIRECT POINTER
000104     321                          .BYTE   UC.PWF+UC.NPR+UC.ALG+1  ;U.CTL,
000105     000                          .BYTE   0       ;U.STS, UNIT STATUS
000106     001                          .BYTE   1       ;U.UNIT, UNIT NUMBER
000107     000                          .BYTE   0       ;U.ST2, UNIT STATUS
000110  000000                          .WORD   0       ;U.CW1
000112  000000                          .WORD   0       ;U.CW2
000114  000000                          .WORD   0       ;U.CW3, BLOCK COUNT FOR MULTI-BLOCK DMA IO
000116  000000                          .WORD   0       ;U.CW4
000120  000162'                         .WORD   ACSCB1  ;U.SCB, POINTER TO UNIT'S SCB
000122  000000                          .WORD   0       ;U.ATT
000124  000000                          .WORD   0       ;U.BUF
000126  000000                          .WORD   0       ;U.CNT
000130  000000                          .WORD   0       ;
                                ;
                                ; STATUS CONTROL BLOCK --- UNIT 0
                                ;
000132                          ACSCB0::
000132  000000                          .WORD   0       ;S.LHD
000134  000132'                         .WORD   .-2
000136     240                          .BYTE   PR5
000137     036                          .BYTE   170/4   ;S.VCT, DEVICE INTERRUPT VECTOR/4
000140     000                          .BYTE   0       ;S.CTM, CURRENT TIME OUT COUNT
000141     024                          .BYTE   20.     ;S.ITM, INITIAL TIME OUT COUNT
000142     000                          .BYTE   0       ;S.CON, CONTROLLER INDEX TIMES 2
000143     000                          .BYTE   0       ;S.STS, CONTROLLER STATUS
000144  167640                          .WORD   167640  ;S.CSR, ADDRESS OF THE FIRST CSR
000146  000000                          .WORD   0       ;S.PKT, POINTER TO THE ACTIVE I/O PACKET
000150                                  .BLKW   5       ;S.FRK, FORK SAVE AREA
                                ;
                                ; STATUS CONTROL BLOCK --- UNIT 1
                                ;
000162                          ACSCB1::
000162  000000                          .WORD   0       ;S.LHD
000164  000162'                         .WORD   .-2
000166     240                          .BYTE   PR5
000167     036                          .BYTE   170/4   ;S.VCT, DEVICE INTERRUPT VECTOR/4
000170     000                          .BYTE   0       ;S.CTM, CURRENT TIME OUT COUNT
000171     024                          .BYTE   20.     ;S.ITM, INITIAL TIME OUT COUNT
000172     002                          .BYTE   2       ;S.CON, CONTROLLER INDEX TIMES 2
000173     000                          .BYTE   0       ;S.STS, CONTROLLER STATUS
000174  167640                          .WORD   167640  ;S.CSR, ADDRESS OF THE FIRST CSR
000176  000000                          .WORD   0       ;S.PKT, POINTER TO THE ACTIVE I/O PACKET
000200                                  .BLKW   5       ;S.FRK, FORK SAVE AREA
                                ;
                                ;
                                ; END OF THE ACC HANDLER DATA BASE
                                ;
000212                          $ACEND::
        000001                          .END
```

```
ACDCB·   000000R·
ACEND·   000074R·
ACSCB0   000132RG·
ACSCB1   000162RG·
ACSTRT·  000036R·
A$$CHK=·000000
A$$CPS=·000000
A$$C11=·000002·
A$$NSI=·000000
A$$PRI=·000000
A$$TRP=·000000
CMODE·=·140000
C$$CDA=·000002·
C$$CKP=·000000
C$$CSR=·177404
C$$INT=·000000
C$$ORE=·002000
C$$RSH=·177564
C$$RUN=·000000
C$$TTY=·177564
DV.CCL=·000002·
DV.COM=·020000
DV.DIR=·000010
DV.F11=·040000
DV.ISP=·002000
DV.MNT=·100000
DV.MXD=·000100
DV.OSP=·004000
DV.PSE=·010000
DV.REC=·000001
DV.SDI=·000020
DV.SQD=·000040
DV.SWL=·001000
DV.TTY=·000004
DV.UMD=·000200
D$$ISK=·000000
D$$L11=·000004
D$$SHF=·000000
D$$SYNC=·000000
D$$YNM=·000000
D.DSP·   000012·
D.LNK·   000000
D.MSK·   000014
D.NAM·   000004
D.PCB·   000034
D.UCB·   000002·
D.UCBL·  000010
D.UNIT·  000006
D.VCAN=·000002·
D.VINI=·000000
D.VOUT=·000004
D.VPWF=·000006
E$$XPR=·000000
FE.CAL=·000040
FE.CEX=·020000
FE.DRV=·000010
FE.EXP=·000200
FE.EXT=·000001
FE.EXV=·000004
FE.LSI=·000400
FE.MUP=·000002·
FE.MXT=·040000
FE.NLG=·100000
FE.PKT=·000100
FE.PLA=·000020
F$$LVL=·000001
G$$TPP=·000000
G$$TSS=·000000
G$$TTK=·000000
G$$WRD=·000000
H$$RTZ=·000074
I$$RAR=·000000
I$$RDN=·000000
KDSAR0=·172360
KDSDR0=·172320
KISAR0=·172340
KISAR5=·172352·
KISAR6=·172354
KISAR7=·172356
KISDR0=·172300
KISDR6=·172314
KISDR7=·172316
K$$CNT=·177546
K$$CSR=·177546
K$$IEN=·000115
K$$LDC=·000001
K$$TPS=·000074
LD$XM·=·000000
L$$ASG=·000000
L$$DRV=·000000
L$$PTO=·000144
L$$P11=·000001
L.ASG·   000010
L.LGTH=·000012·
L.LNK·   000000
L.NAM·   000002·
L.TYPE   000005
L.UCB·   000006
L.UNIT   000004
MPAR·  =·172100
MPCSR·=·177746
M$$CRB=·000124
M$$CRX=·000000
M$$FCS=·000000
M$$MGE=·000000
M$$MUP=·000000
M$$OVR=·000000
M.BFVH   000011
M.BFVL   000012·
M.LGTH=·000014
M.LNK·   000000
M.UMRA·  000002·
M.UMRN·  000004
M.UMVH·  000010
M.UMVL·  000006
N$$LDV=·000001
N$$MOV=·000041
PIRQ·  =·177772·
PMODE·=·030000
PR0    =·000000
PR1    =·000040
PR4    =·000200
PR5    =·000240
PR6    =·000300
PR7    =·000340
PS     =·177776
P$$GMX=·000000
P$$LAS=·000000
P$$P45=·000000
P$$RFL=·000000
P$$RTY=·000000
P$$SRF=·000000
P$$WRD=·000000
Q$$OPT=·000007
R$$DER=·000000
R$$EXV=·000000
R$$K11=·000001
R$$P11=·000001
R$$SND=·000000
R$$11M=·000000
SISDR0=·172200
SPARE·=·000010
SP.EIP=·000001
SP.ENB=·000002·
SP.LOG=·000004
SR0    =·177572·
SR3    =·172516
SWR·   =·177570
S$$YSZ=·007600
S.BMSK·  177776
S.BMSV·  177774
S.CON·   000010
S.CSR·   000012·
S.CTM·   000006
S.FRK·   000016
S.ITM·   000007
S.LHD·   000000
S.PKT·   000014
S.PRI·   000004
S.RCNT·  177772·
S.ROFF·  177773
S.STS·   000011
S.VCT·   000005
TPS·   =·177564
T$$ACR=·000000
T$$BTW=·000000
T$$BUF=·000000
T$$CCA=·000000
T$$CCO=·000000
T$$CTR=·000000
T$$GMC=·000000
T$$GTS=·000000
T$$KMG=·000000
T$$LWC=·000000
T$$M11=·000001
T$$RNE=·000000
T$$RPR=·000000
T$$RST=·000000
T$$RUB=·000000
T$$SYN=·000000
T$$TRW=·000000
T$$UTB=·000000
T$$VBF=·000000
T$$30P=·000000
UBMPR·=·170200
UC.ALG=·000200
UC.ATT=·000010
UC.KIL=·000004
UC.LGH=·000003
UC.NPR=·000100
UC.PWF=·000020
UC.QUE=·000040
UDSAR0=·177660
UDSDR0=·177620
UISAR0=·177640
UISAR4=·177650
UISAR5=·177652·
UISAR6=·177654
UISAR7=·177656
UISDR0=·177600
UISDR4=·177610
UISDR5=·177612·
UISDR6=·177614
UISDR7=·177616
US.ABO=·000001
US.BSP=·000002·
US.BSY=·000200
US.CRW=·000004
US.DSB=·000010
US.ECH=·000002·
US.FOR=·000040
US.FRK=·000002·
US.LAB=·000004
US.MDE=·000002·
US.MDM=·000020
US.MNT=·000100
US.OFL=·000001
US.OUT=·000001
US.PUB=·000004
US.PWF=·000010
US.RED=·000002
US.SHR=·000001
US.SPU=·000002
US.UMD=·000010
US.WCK=·000010
U.ACP·=·000032
U.ATT·   000022
U.BUF·   000024
U.CBF·=·000032
U.CNT·   000030
U.CTL·   000004
U.CW1    000010
U.CW2·   000012
U.CW3    000014
U.CW4    000016
U.DCB·   000000
U.LUIC·  177774
U.OWN·   177776
U.RED·   000002
U.SCB·   000020
U.STS·   000005
U.ST2·   000007
U.UIC·=·000052
U.UNIT=·000006
U.VCB·=·000034
U2.AT.=·000020
U2.CRT=·002000
U2.DHI=·100000
U2.DJI=·040000
U2.DZ1=·000100
U2.ESC=·001000
U2.HLD=·000040
U2.LOG=·000400
U2.LWC=·000001
U2.L3S=·000004
U2.L8S=·010000
U2.NEC=·004000
U2.PRV=·000010
U2.RMT=·020000
U2.R04=·100000
U2.SLV=·000200
U2.VT5=·000002
U2.7CH=·010000
V$$CTR=·000404
V$$RSN=·000031
X$$DBT=·000000
X$$M11=·000002
$ACDAT·  000000RG·
$ACEND=  000212RG·
.AG0·=·  000042RG·
.AC1     000100RG·

. ABS.  177776     000
```

```
        000212        001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3639 WORDS  ( 15 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:12
ACTAB,ACTAB/-SP=[1,1]EXEMC/ML,[200,200]RSXMC/PA:1,[20,1]ACTAB
```

10- 2 MACRO'S AND CONSTANTS
11- 25 ASSEMBLY-TIME DATA DEFINITIONS
12- 103 INIT THE DMC - READ HANG LOOP
13- 142 SPOOL FILES
14- 168 SINGLE BLOCK EXCHANGES
15- 197 DHR BUFFERS
16- 274 SUBROUTINES

```
                                .TITLE  DMCIN
                                .SBTTL  MACRO'S AND CONSTANTS
                        ;
                                .MCALL  QIOW$C,QIOW$S,EXIT$S
                                .MCALL  SDAT$C,WTSE$S,CLEF$S
                                .MCALL  RSUM$C
                                .MCALL  FINIT$,FDAT$R,OFNB$W,WRITE$,WAIT$,CLOSE$
                                .MCALL  FDOF$L,FCSBT$,FDBDF$,FDRC$A,FDOP$A,FSRSZ$
                                .MCALL  FDBK$A
                        ;
                                .GLOBL  BLDNFL,,DLFNB
                                .GLOBL  SUINDX,SRECPT,GETFRE,PUTSSQ
                                .GLOBL  DHRCOM,SUDHRI,SUST
                                .GLOBL  SN
                        ;
                        ;LUNS
000001                  DPLUN=1
000002                  XMLUN=2
000006                  COLUN=6
                        ;
                        ;MISC EQUATES
000001                  EF,IO=1         ;I/O EVENT FLAG
                        ;
```

```
                                                .SBTTL  ASSEMBLY-TIME·DATA·DEFINITIONS·
000000                                          .PSECT·
                                        ;
                                        ;I/O·STATUS·BLOCKS·
000000                                  DMCIOS: .BLKW·  2·
000004                                  SYIOS:  .BLKW·  2·
                                        ;
                                        ;
                                        ;
000010                                          FDOF$L·
000010                                          FCSBT$
                                        ;
                                        ;OUTPUT·FILE·FDB·
000010                                  OFILE:  FDBDF$
000150                                          FDRC$A· FD.RWM·
000150                                          FDBK$A· .N.BUFB,.EF.IO.SYIOS·
000150                                          FDOP$A· DPLUN·
                                        ;
000150                                          FSRSZ$  0
                                        ;
                                        ;EXCHANGE·ID·PARSE·TABLE·
000150  117     106                     EXCHID: .ASCII· /OF/
000152· 000640'                                 .WORD·  FOSRCV·
000154  101     104                             .ASCII· /AD/
000156  000702'                                 .WORD·  DUBSPL·
000160  124     122                             .ASCII· /TR/
000162· 000702'                                 .WORD·  DUBSPL·
000164  110     104                             .ASCII· /HD/
000166  001030'                                 .WORD·  DHRBUF·
000170  130     123                             .ASCII· /XS/
000172· 000736'                                 .WORD·  SUXDON·
000174  104     125                             .ASCII· /DU/
000176  000752'                                 .WORD·  DTBEND·
000200  122·    103                             .ASCII· /RC/
000202· 000766'                                 .WORD·  STATRC·
        000007                          TABLSZ=<.-EXCHID>/4
                                        ;
                                        ;ERROR·MESSAGES·
000204  000034                          ERR1:   .WORD·  ERR1L·
000206  104     115     103             ERR1T:  .ASCII· /DMCINX: DMC·START-UP·FAILURE/·
000211  111     116     130
000214  072·    040     104
000217  115     103     040
000222· 123     124     101
000225  122·    124     055
000230  125     120     040
000233  106     101     111
000236  114     125     122·
000241  105
        000034                          ERR1L=.-ERR1T·
                                                .EVEN·
                                        ;
000242· 000030                          ERR2:   .WORD·  ERR2L·
000244  104     115     103             ERR2T:  .ASCII· /DMCINX: DMC·READ·FAILURE/·
000247  111     116     130
000252· 072·    040     104
000255  115     103     040
```

```
                000260     122     105     101
                000263     104     040     106
                000266     101     111     114
                000271     125     122     105
     70         000030                     ERR2L=.-ERR2T
     71                                            .EVEN
     72
     73 000274  000041                     ERR3:   .WORD   ERR3L
     74 000276      104     115     103    ERR3T:  .ASCII  /DMCINX: UNKNOWN EXCHANGE RECEIVED/
        000301      111     116     130
        000304      072     040     125
        000307      116     113     116
        000312      117     127     116
        000315      040     105     130
        000320      103     110     101
        000323      116     107     105
        000326      040     122     105
        000331      103     105     111
        000334      126     105     104
     75         000041                     ERR3L=.-ERR3T
     76                                            .EVEN
     77                                    ;
     78 000340  000022                     ERR4:   .WORD   ERR4L
     79 000342      104     115     103    ERR4T:  .ASCII  /DMCINX: DISK ERROR/
        000345      111     116     130
        000350      072     040     104
        000353      111     123     113
        000356      040     105     122
        000361      122     117     122
     80         000022                     ERR4L=.-ERR4T
     81                                            .EVEN
     82
     83 000364  000030                     ERR5:   .WORD   ERR5L
     84 000366      104     115     103    ERR5T:  .ASCII  /DMCINX:         LINK INL         /
        000371      111     116     130
        000374      072     011     114
        000377      111     116     113
        000402      040     111     116
        000405      114     040     040
        000410      040     040     040
        000413      040     040     040
     85         000030                     ERR5L=.-ERR5T
     86                                            .EVEN
     87                                    ;
     88 000416  000026                     ERR6:   .WORD   ERR6L
     89 000420      104     115     103    ERR6T:  .ASCII  /DMCINX: CAN'T RUN TASK/
        000423      111     116     130
        000426      072     040     103
        000431      101     116     047
        000434      124     040     122
        000437      125     116     040
        000442      124     101     123
        000445      113
     90         000026                     ERR6L=.-ERR6T
     91                                            .EVEN
     92                                    ;
     93                                    ;
```

```
                                        ;MISC LOCATIONS
000446  000000                          BLKCNT: .WORD   0       ;EXCHANGE BLOCK COUNT
000450                                  SDATA:  .BLKW   13.     ;SEND DATA BUFFER
000502  000000                          UNITNB: .WORD   0       ;THIS DMCIN'S DMC UNIT NUMBER*2 (0,2,4,...)
000504  000000                          ADHRCT: .WORD   0       ;ADDRESS OF THIS DMCIN'S DHR CONTROL WORDS
                                        ;
000506  000001  000002                  BITMAP: .WORD   BIT0,BIT1
                                        ;
```

```
                                        .SBTTL  INIT THE DMC - READ HANG LOOP
                                ;
000512                          DMCIN:
000512                                  FINIT$
                                ;
                                ; DETERMINE OUR UNIT NUMBER
000516  012705  000000G                 MOV     #SN,R5                  ;SAVE OUR UNIT # TIMES 2
000522  060505                          ADD     R5,R5
000524  010567  177752                  MOV     R5,UNITNB
000530  016505  000000G                 MOV     SUDHRI(R5),R5           ;CONTROL TABLE ADDRESS
000534  010567  177744                  MOV     R5,ADHRCT               ;SAVE IT
000540  016505  000002                  MOV     DH.BF0(R5),R5           ;FIRST BUFFER ADDRESS
000544  062705  000000G                 ADD     #DHRCOM,R5
                                ;
                                ;INIT DMC - WAIT FOR CONTROL
000550                                  QIOW$C  IO.TRM,XMLUN,EF,IO      ;TERMINATE LINK
000556                          INLDMC: QIOW$C  IO.INL,XMLUN,EF,IO      ;INIT LINK
000564  012700  000364'                 MOV     #ERR5,R0                ;MSG TO CONSOLE -
000570                                  CALL    COOUT                   ;  LINK INITIALIZED
                                ;
                                ;
                                ;HANG A READ FOR THE NEXT EXCHANGE
000574                          HANGRD: CALL    READMC                  ;READ LINK
000600  103766                          BCS     INLDMC                  ;ERROR, RE-INITIALIZE
                                ;
                                ;PARSE EXCHANGE ID
000602  012701  000150'         MSTRST: MOV     #EXCHID,R1              ;PARSE TABLE ADDRESS
000606  012700  000007                  MOV     #TABLSZ,R0              ;TABLE SIZE
000612  022115                  1$:     CMP     (R1)+,(R5)              ;CHECK ID MATCH
000614  001001                          BNE     2$                      ;NO
000616  000131                          JMP     @(R1)+                  ;GOT MATCH - GO TO ROUTINE
                                ;
000620  062701  000002          2$:     ADD     #2,R1                   ;NEXT ENTRY
000624  077006                          SOB     R0,1$                   ;LOOP
000626  012700  000274'                 MOV     #ERR3,R0                ;YES - REPORT ERROR
000632                                  CALL    COOUT
000636  000756                          BR      HANGRD
                                ;
```

```
                                        .SBTTL  SPOOL·FILES·
                                ;
                                ; GOT·FOS·IN·
000640  016704  177636          FOSRCV: MOV·    UNITNB,R4               ;UNIT·#·OF OUR·DMC·
000644  016404  000000G·                MOV·    SUINDX(R4),R4           ;PUT·FDSC·IN·SU·
000650  062704  000002                  ADD·    #SS,FID,R4              ;  STATUS·TABLE·
000654  016401  000006                  MOV·    FD,FNB(R4),R1           ;FOS·FILE·NUMBER·
000660                                  CALL·   WRTFIL·                 ;DMC·DATA·TO·FILE·
000664  103734                          BCS·    INLDMC·                 ;RETRY·EXCHANGE·
                                ;
000666  012700  000001                  MOV·    #1,R0                   ;SSQ·COMMAND·CODE·
000672·                                 CALL·   GOSCHD·                 ;TELL·SCHED·FOS·IS·IN·
000676  000167  177672                  JMP·    HANGRD·
                                ;
                                ;
                                ; RETRIEVED·DOCUMENT OR·DATA·BASE·UPDATE SPOOL·FILE·
000702· 012704  000452'         DUBSPL: MOV·    #SDATA+2,R4             ;PUT·FDSC·IN·SEND·PACKET·
000706  012701  000026                  MOV·    #FN.DBR,R1              ;FILE·NUMBER·
000712·                                 CALL·   WRTFIL·                 ;DMC·DATA·TO·FILE·
000716  103717                          BCS·    INLDMC·                 ;RETRY·EXCHANGE·
                                ;
000720  112767  000004  177523          MOVB·   #4,SDATA+1              ;COMMAND·BYTE·
000726                                  CALL·   GOHOTK·                 ;QUEUE·TO·HOTSK·
000732· 000167  177636                  JMP·    HANGRD·
                                ;
```

```
                                        .SBTTL  ··SINGLE·BLOCK·EXCHANGES·
                                ;
                                ;
                                ; SUX%·DONE·
000736  012700  000000          SUXDON: MOV·    #0,R0                   ;COMMAND·0 TO·MSCHED
000742·                                 CALL·   GOSCHD·
000746  000167  177622                  JMP·    HANGRD·
                                ;
                                ;
                                ; DATA·BASE·END·
000752·012700  000002           DTBEND: MOV·    #2,R0                   ;COMMAND·2 TO·MSCHED
000756                                  CALL·   GOSCHD·
000762·000167  177606                   JMP·    HANGRD·
                                ;
                                ;
                                ; STATUS·RECORD·- COPY·TO·MCOM
000766  016700  177510          STATRC: MOV·    UNITNB,R0               ;GET·ADDRESS·OF·THIS SU'S·
000772·016000  000000G·                 MOV·    SRECPT(R0),R0           ;  STATUS·RECORD·IN·MCOM·
000776  010501                          MOV·    R5,R1                   ;INPUT·DATA·ADDRESS·
001000  062701  000002                  ADD·    #2,R1
001004  012702·000051                   MOV·    #SR.LEN/2,R2·           ;LENGTH·OF DATA·
001010  012120                  1$:     MOV·    (R1)+,(R0)+             ;COPY·DATA TO·MCOM·
001012·077202·                          SOB·    R2,1$
                                ;
001014  012700  000003                  MOV·    #3,R0                   ;COMMAND·3 TO·MSCHED
001020                                  CALL·   GOSCHD·
001024  000167  177544                  JMP·    HANGRD·
                                ;
```

```
                                        .SBTTL··DHR·BUFFERS·
                                ;
                                ; GOT·A·DHR·BUFFER·INPUT·- NOT CONCATENATING·YET·
                                ;
001030  005765  003776          DHRBUF: TST·    N.BUFB-2(R5)            ;BRANCH·IF CONTINUATION·DHR·
001034  100467                          BMI·    STRTCT·
                                ;
                                ;
                                ; SINGLE-BLOCK·DHR·
001036  116765  177776G·000002·        MOVB·   SUST-2,2(R5)            ;BATCH·#·IN·DHR·BUFFER·
001044  105065  000003                  CLRB·   3(R5)
                                ;
001050  016702· 177430                  MOV·    ADHRCT,R2·              ;CONTROL·AREA·ADDRESS·
001054  116203  000010                  MOVB·   DH.DMC(R2),R3           ;CURRENT·BUFFER·#·
001060  042703  177776                  BIC·    #177776,R3
001064  060303                          ADD·    R3,R3
001066  056362· 000506' 000000          BIS·    BITMAP(R3),DH.CTL(R2)   ;SHOW·ITS·FULL·
                                ; QUEUE·BUFFER·TO·HOTSK·
001074  112767  000005  177347          MOVB·   #5,SDATA+1              ;BUILD·SEND·PACKET·
001102· 010267  177344                  MOV·    R2,SDATA+2·
001106  010567  177342                  MOV·    R5,SDATA+4
001112· 016367  000506' 177336          MOV·    BITMAP(R3),SDATA+6
001120                                  CALL·   GOHOTK·                 ;CALL·HOTSK·
                                ;
001124  105262· 000010                  INCB·   DH.DMC(R2)              ;NEXT·BUFFER·NUMBER·
001130  062703  000002                  ADD·    #2,R3
001134  042703  177775                  BIC·    #177775,R3              ;NEXT·BUFFER·INDEX·
001140  036362· 000506' 000000  TNXTBF: BIT·    BITMAP(R3),DH.CTL(R2)   ;SEE·IF·NEXT·BUFFER·BUSY·
001146  001007                          BNE·    WAITBF·                 ;YES·
001150  060203                          ADD·    R2,R3                   ;NEXT·BUFFER·ADDRESS
001152· 016305  000002                  MOV·    DH.BF0(R3),R5
001156· 062705  000000G·                ADD·    #DHRCOM,R5
001162· 000167  177406                  JMP·    HANGRD·
                                ; WAIT·FOR·HOTSK·TO·FREE·NEXT·BUFFER·
001166                          WAITBF: WTSE$S·  DH.FLG(R2)             ;WAIT·
001200                                  CLEF$S·  DH.FLG(R2)
001212· 000752·                         BR      TNXTBF·                 ;TRY·BUFFER·AGAIN·
                                ;
                                ;
                                ; START·OF·DHR·CONTINUATION·RECORDS·
001214  012703  000010          STRTCT: MOV·    #2*N.BFAC,R3            ;OPEN·OUTPUT·FILE·- MIN·2·BLOCKS·
001220  012701  000040                  MOV·    #FN.DHR,R1              ;FILE·NUMBER·
001224                                  CALL·   OPENFL·
001230  012715  046504          NEXTHD: MOV·    #"DM,(R5)               ;EXCHANGE·ID·
001234  116765  177776G·000002·        MOVB·   SUST-2,2(R5)            ;BATCH·NUMBER·
001242· 105065  000003                  CLRB·   3(R5)
001246                                  CALL·   WRTBLK·                 ;WRITE·A·BLOCK·
001252· 005765  003776                  TST·    N.BUFB-2(R5)            ;BRANCH·IF LAST·BLOCK·
001256  001432·                         BEQ·    DONCNT·
                                ;
001260                          NEXTBF: CALL·   READMC·                 ;READ·NEXT BUFFER·
001264  103004                          BCC·    1$                      ;NO·ERROR·
001266                                  CALL·   DLETFL·                 ;ERROR·- DELETE·FILE
001272· 000167  177260                  JMP·    INLDMC·                 ;RETRY·EXCHANGE·
                                ;
001276· 022715  051530          1$:     CMP·    #"XS,(R5)               ;BRANCH·IF NOT·SUX%·DONE·
001302· 001005                          BNE·    2$
```

```
    254 001304  012700  000000                  MOV·    #0,R0                   ;SUX%·DONE ALLOWED·TO·INTERLEAVE·
    255 001310                                  CALL·   GOSCHD·
    256 001314  000761                          BR      NEXTBF·                 ;GET·NEXT·BUFFER·
    257                                 ;
    258·001316  022715  042110          2$:     CMP·    #"HD,(R5)               ;WRITE·BUFFER·TO·FILE·IF·
    259 001322· 001742·                         BEQ·    NEXTHD·                 ;  DHR·BUFFER·
    260 001324                                  CALL·   DLETFL·                 ;SYNC·ERROR·- ABORT·
    261 001330  012700  000274'                 MOV·    #ERR3,R0                ;UNKNOWN·EXCHANGE·
    262·001334                                  CALL·   COOUT·
    263 001340  000167  177236                  JMP·    MSTRST·                 ;PARSE·EXCHANGE·
    264                                 ; DONE·WITH·CONTINUATION·RECORDS·
    265 001344  012704  000452'         DONCNT: MOV·    #SDATA+2,R4             ;BUILD·FDSC·IN·SEND·PACKET·
    266 001350  012701  000040                  MOV·    #FN.DHR,R1              ;FILE·NUMBER·
    267 001354                                  CALL·   CLOSFL·                 ;CLOSE·OUTPUT·FILE·
    268 001360  112767  000006  177063          MOVB·   #6,SDATA+1              ;COMMAND·TO·HOTSK·
    269 001366                                  CALL·   GOHOTK·
    270 001372· 000167  177176                  JMP·    HANGRD·
    271                                 ;
    272·                                ;
```

```
                                        .SBTTL  SUBROUTINES
                                ;
                                ;
                                ; GO TO SCHEDULER IN SSQ
                                ; INPUT - R0: COMMAND BYTE
                                ;
001376                          GOSCHD: CALL    GETFRE                  ;GET A PACKET
001402  112762  000006  000002          MOVB    #XDMCIN,2(R2)           ;SOURCE ID
001410  110062  000003                  MOVB    R0,3(R2)                ;COMMAND
001414  016762  177062  000004          MOV     UNITNB,4(R2)            ;SU NUMBER
001422  006262  000004                  ASR     4(R2)
001426                                  CALL    PUTSSQ                  ;PUT PACKET TO SCHED
001432  000207                          RTS     PC
                                ;
                                ;
                                ; GO TO HOTSK IN RCVQ
                                ; INPUT - SDATA+1: COMMAND BYTE
                                ;         SDATA+2: PACKET DATA
                                ;
001434  112767  000006  177006  GOHOTK: MOVB    #XDMCIN,SDATA           ;COMMAND SOURCE
001442                                  SDAT$C  HOTSK,SDATA             ;SEND DATA TO HOTSK
001450  103004                          BCC     TSKGD
001452  012700  000416'          TSKERR: MOV     #ERR6,R0                ;ERROR
001456  000167  000552                  JMP     ERROR
                                ;
001462                          TSKGD:  RSUM$C  HOTSK                   ;RESUME HOTSK
001470  103004                          BCC     1$                      ;BRANCH IF OK
001472  026727  000000G 000000G        CMP     $DSW,#IE.ITS            ;IE.ITS ONLY ACCEPTABLE ERROR
001500  001364                          BNE     TSKERR
001502  000207                  1$:     RTS     PC
                                ;
                                ;
                                ; READ THE DMC - INTO BUFFER 0 OR 1
                                ; INPUT/OUTPUT - R5= BUFFER ADDRESS
                                ;
001504                          READMC: QIOW$S  #IO.RLB,#XMLUN,#EF,IO,,#DMCIOS,,<R5,#N.BUFB>
001554  122767  000000G 176216          CMPB    #IS.SUC,DMCIOS          ;SUCCESS?
001562  001424                          BEQ     1$                      ;YES
001564  122767  000000G 176206          CMPB    #IE.CNR,DMCIOS          ;CONNECTION REJECTED?
001572  001422                          BEQ     2$                      ;YES, RETURN INIT REQUEST
001574  122767  000000G 176176          CMPB    #IE.DNR,DMCIOS          ;DEVICE NOT READY?
001602  001416                          BEQ     2$                      ;YES, RETURN INIT REQUEST
001604  122767  000000G 176166          CMPB    #IE.ABO,DMCIOS          ;REQUEST ABORTED?
001612  001412                          BEQ     2$                      ;YES, RETURN INIT REQUEST
001614  122767  000000G 176156          CMPB    #IE.TMO,DMCIOS          ;TIME OUT?
001622  001406                          BEQ     2$                      ;YES
001624  012700  000242'                  MOV     #ERR2,R0                ;NO - ERROR EXIT
001630  000167  000400                  JMP     ERROR
                                ;
001634                          1$:
001634  000241                          CLC
001636  000207                          RTS     PC
001640                          2$:     SAVE    R0,R1,R2
001646  012700  000410'                  MOV     #<ERR5T+22>,R0  ;R0->OUTPUT STRING
001652  116701  176122                  MOVB    DMCIOS,R1       ;R1 = ERROR NUMBER
001656  012702  000001                  MOV     #1,R2           ;DON'T SUPPRESS LEADING ZEROS
001662                                  CALL    $CBDSG
```

```
                                        RESTOR·  R0,R1,R2·
                                        SEC·                            ;REQUEST·RE-INITIALIZATION·
                                        RTS·     PC·                    ;RETURN·
                                ;
                                ;
                                ; WRITE·A·SPOOL·FILE
                                ; INPUT: R4- ADDRESS OF·WHERE·TO·BUILD·FDSC·
                                ;         R5- ADDRESS OF·FIRST·BUFFER·
                                ;         R1- FILE·NUMBER·
                                ;
001700  016503  000002          WRTFIL: MOV·     2(R5),R3               ;EXCHANGE·BLOCK·COUNT·
001704  010367  176536                  MOV·     R3,BLKCNT·
001710  070327  000004                  MUL·     #N,BFAC,R3             ;#·OF·VIRTUAL·BLOCKS
001714                                  CALL·    OPENFL·
                                ;
001720                          2$:     CALL·    WRTBLK·                ;WRITE·BLOCK·TO·FILE
001724  005367  176516                  DEC·     BLKCNT·                ;BRANCH·IF·DONE·
001730  001407                          BEQ·     3$
001732·                                 CALL·    READMC·                ;NEXT·EXCHANGE·BLOCK
001736  103370                          BCC·     2$                     ;NO·ERROR·
001740                                  CALL·    DLETFL·                ;ERROR·-·DELETE·FILE
001744  000261                          SEC·                            ;RETRY·EXCHANGE·
001746  000207                          RTS·     PC·
                                ;
001750                          3$:     CALL·    CLOSFL·                ;CLOSE·FILE·
001754  000241                          CLC·
001756  000207                          RTS·     PC·
                                ;
                                ;
                                ; OPEN·OUTPUT·FILE·
                                ; INPUT: R3- #·OF·VIRTUAL·BLOCKS·TO·ALLOCATE·
                                ;         R1- FILE·NUMBER·
                                ;
001760  005403                  OPENFL:·NEG·     R3                     ;INITIAL·ALLOCATION·
001762·                                 FDAT$R·  #OFILE,,,,R3
001772·                                 CALL·    BLDNFL·                ;BUILD·FNB
001776                                  OFNB$W·                         ;OPEN·
002010  103445                          BCS·     FILERR·                ;ERROR·
002012·  000207                         RTS·     PC·
                                ;
                                ; WRITE·A·BLOCK·TO·THE·FILE·
                                ; INPUT: R5- BUFFER·ADDRESS·
                                ;
002014                          WRTBLK: WRITE$   #OFILE,R5              ;WRITE·
002030                                  WAIT$
002034  122767  000000G·175742·        CMPB·    #IS.SUC,$YIOS·         ;TEST·SUCCESS·
002042·  001030                         BNE·     FILERR·
002044  000207                         RTS·     PC·
                                ;
                                ; CLOSE·FILE·-·CREATE·FDSC·
                                ; INPUT: R4- ADDRESS·WHERE·TO·BUILD·FDSC
                                ;         R1- FILE·NUMBER·
                                ;
002046  012700  000010'         CLOSFL: MOV·     #OFILE,R0              ;FDB·
002052·  016064  000102  000000         MOV·     F.FNB+N.FID(R0),FD.FID(R4) ;BUILD·FDSC·
002060  016064  000104  000002·         MOV·     F.FNB+N.FID+2(R0),FD.FID+2(R4)
002066  016064  000120  000004         MOV·     F.FNB+N.FVER(R0),FD.FVR(R4)
```

```
 388 002074  010164  000006                    MOV·     R1,FD.FNB(R4)
 389 002100                                     CLOSE$                          ;CLOSE·
 390 002104  103407                             BCS·     FILERR·
 391 002106  000207                             RTS·     PC·
 392·                                  ;
 393                                   ; DELETE·THE·FILE·
 394                                   ;
 395 002110  012700  000010'           DLETFL: MOV·     #OFILE,R0              ;FDB·
 396 002114                                     CALL·    .DLFNB·                ;DELETE·
 397 002120  103401                             BCS·     FILERR·
 398 002122· 000207                             RTS·     PC·
 399                                   ;
 400                                   ;
 401                                   ; ERROR·WRITING·TO·FILE·
 402·                                  ;
 403 002124  012700  000010'           FILERR: MOV·     #OFILE,R0              ;TRY·TO·DELETE·
 404 002130                                     CALL·    .DLFNB·
 405 002134  012700  000340'                    MOV·     #ERR4,R0               ;ERROR·MESSAGE·AND·EXIT·
 406 002140  000435                             BR       ERROR·
 407                                   ;
 408                                   ;
 409                                   ;OUTPUT·TO·CONSOLE·
 410                                   ;
 411                                   ;R0=MESSAGE·BLOCK·ADDRESS·
 412·                                  ;
 413 002142· 012001                    COOUT:  MOV·     (R0)+,R1               ;MESSAGE·ADDRESS·AND·LENGTH·
 414 002144  116760  176332  000005             MOVB·    UNITNB,5(R0)           ;IDENTIFY·DMC·
 415 002152· 106260  000005                     ASRB·    5(R0)
 416 002156  152760  000060  000005            BISB·    #60,5(R0)
 417 002164                                     QIOW$S·  #IO.WLB,#COLUN,#EF,IO,,,,<R0,R1,#40>    ;MSG·TO·CONSOLE·
 418 002232· 000207                             RTS·     PC·
 419                                   ;
 420                                   ;
 421                                   ;ERROR·EXIT·
 422·                                  ;
 423                                   ;R0=MESSAGE·BLOCK·ADDRESS·
 424                                   ;
 425 002234                            ERROR:  CALL·    COOUT·                 ;MESSAGE·TO·CONSOLE·
 426 002240                                     EXIT$S·
 427                                   ;
 428·                                  ;
 429                                   ;
 430         000512'                           .END·    DMCIN·
```

```
ADHRCT  000504R        B.NQRY  000232        010 ERR4T   000342R        FN.DBS  000022        011 F.OVBS=  000030
BITMAP  000506R        B.QLSZ  000106        010 ERR5    000364R        FN.DHR  000040        011 F.RACC=  000016
BITVAL=  000000        B.QMAP  000234        010 ERR5L =  000030        FN.EMA  000012        011 F.RATT=  000001
BIT0   =  000001       B.QSPL  000316        010 ERR5T   000366R        FN.EMB  000014        011 F.RCNM=  000034
BIT1   =  000002       B.QTTM  000076        010 ERR6    000416R        FN.EMC  000016        011 F.RCTL=  000017
BIT10  =  002000       B.QUQP  000056        010 ERR6L =  000026        FN.FSA  000000        011 F.RSIZ=  000002
BIT11  =  004000       B.SFDB  000010        010 ERR6T   000420R        FN.FSB  000002        011 F.RTYP=  000000
BIT12  =  010000       B.SIZE  000772        010 EXCHID  000150R        FN.FSC  000004        011 F.SEQN=  000100
BIT13  =  020000       B.SNDP  000012        010 FA.APD=  000100        FN.LGQ  000034        011 F.SPDV=  000072
BIT14  =  040000       B.SSQ   000004        010 FA.CRE=  000010        FN.LGU  000036        011 F.SPUN=  000074
BIT15  =  100000       B.SSQF  000050        010 FA.DLK=  001000        FN.MFQ  000024        011 F.STBK=  000036
BIT2   =  000004       B.STAT  000044        010 FA.ENB=  100000        FN.MHR  000010        011 F.UNIT=  000136
BIT3   =  000010       B.STTE  000053        010 FA.EXC=  002000        FN.NMB  000044        011 F.URBD=  000020
BIT4   =  000020       B.UDOC  000110        010 FA.EXT=  000004        FN.QLS  000006        011 F.VBN  =  000064
BIT5   =  000040       CF.B0  =  000070          FA.NSP=  000100        FN.QRY  000020        011 F.VBSZ=  000060
BIT6   =  000100       CF.B2  =  000067          FA.POS=  010000        FN.SF0  000030        011 GETFRE=  ****** G
BIT7   =  000200       CF.B4  =  000066          FA.RD  =  000001        FN.SF1  000032        011 GOHOTK  001434R
BIT8   =  000400       CF.B6  =  000065          FA.RWD=  004000        FN.SHD  000042        011 GOSCHD  001376R
BIT9   =  001000       CF.DR0=  000064          FA.SED=  040000        FOSRCV  000640R          HANGRD  000574R
BLDNFL=  ****** G      CF.DR1=  000063          FA.SHR=  000040        FO.APD=  000106          IE.ABO=  ****** GX
BLKCNT  000446R        CH.AND=  000001          FA.TMP=  000020        FO.MFY=  000002          IE.CNR=  ****** GX
BS.CLS=  000002        CLOSFL  002046R          FA.WCK=  020000        FO.RD  =  000001          IE.DNR=  ****** GX
BS.DBU=  000004        COLUN  =  000006          FA.WRT=  000002        FO.UPD=  000006          IE.ITS=  ****** GX
BS.INA=  000000        COOUT   002142R          FD.BLK=  000010        FO.WRT=  000016          IE.TMO=  ****** GX
BS.OPN=  000001        DBSLEN=  000116          FD.CCL=  000002        F.ACTL=  000076          INLDMC  000556R
BS.SRC=  000003        DHRBUF  001030R          FD.COM=  020000        F.ALOC=  000040          IO.INL=  ****** GX
BYTE0  =  000000       DHRCOM=  ****** G        FD.CR  =  000002        F.BBFS=  000862          IO.RLB=  ****** GX
BYTE1  =  000001       DH.BF0  000002        005 FD.DIR=  000010        F.BDB  =  000070          IO.TRM=  ****** GX
BYTE2  =  000002       DH.BF1  000004        005 FD.FID  000000    003 F.BGBC=  000057          IO.ULB=  ****** GX
BYTE3  =  000003       DH.CTL  000000        005 FD.FNB  000006    003 F.BKDN=  000026          IS.SUC=  ****** GX
BYTE4  =  000004       DH.DMC  000010        005 FD.FTN=  000001          F.BKDS=  000020          M      =  000062
BYTE5  =  000005       DH.FLG  000006        005 FD.FVR  000004    003 F.BKEF=  000050          MSTRST  000602R
BYTE6  =  000006       DLETFL  002110R             FD.F11=  040000          F.BKP1=  000051          N      =  000002
BYTE7  =  000007       DMCIN   000512R             FD.INS=  000010          F.BKST=  000024          NB.DEV=  000200
BYTE8  =  000010       DMCIOS  000000R             FD.ISP=  002000          F.BKVB=  000064          NB.DIR=  000100
BYTE9  =  000011       DN.DCK  000000        013 FD.LEN  000010    003 F.CHR  =  000075          NB.NAM=  000004
BYTVAL=  000012        DN.NTP  000004        013 FD.MNT=  100000          F.CNTG=  000034          NB.SD1=  000400
B.BSTA  000054    010  DN.NXT  000006        013 FD.OSP=  004000          F.DFNB=  000046          NB.SD2=  001000
B.CNTX  000046    010  DN.ROT  000002        013 FD.PLC=  000004          F.DSPT=  000044          NB.SNM=  000040
B.CQUO  000060    010  DN.SIZ  000010        013 FD.PRN=  000004          F.DVNM=  000134          NB.STP=  000020
B.FEMA  000132    010  DONCNT  001344R             FD.PSE=  010000          F.EFBK=  000010          NB.SVR=  000010
B.FEMB  000142    010  DPLUN  =  000001             FD.RAH=  000001          F.EFN  =  000050          NB.TYP=  000002
B.FEMC  000152    010  DTBEND  000752R             FD.RAN=  000002          F.EOBB=  000032          NB.VER=  000001
B.FFSA  000202    010  DUBSPL  000702R             FD.REC=  000001          F.ERR  =  000052          NEXTBF  001260R
B.FFSB  000212    010  EF.IO  =  000001             FD.RWM=  000001          F.FACC=  000043          NEXTHD  001230R
B.FFSC  000222    010  ERROR   002234R             FD.SDI=  000020          F.FFBY=  000014          N.BFAC=  000004
B.FMHR  000172    010  ERR1    000204R             FD.SOD=  000040          F.FNAM=  000110          N.BHGH=  000006
B.FQLS  000162    010  ERR1L =  000034             FD.TTY=  000004          F.FNB  =  000102          N.BTCH=  000004
B.FSAZ  000100    010  ERR1T   000206R             FD.WBH=  000002          F.FTYP=  000116          N.BUFB=  004000
B.FSBZ  000102    010  ERR2    000242R             FF.CHR=  000005          F.FVER=  000120          N.BUFW=  002000
B.FSCZ  000104    010  ERR2L =  000030             FF.NV  =  000003          F.HIBK=  000004          N.DID  =  000024
B.HBLK  000120    010  ERR2T   000244R             FF.POE=  000002          F.LUN  =  000042          N.DVNM=  000032
B.HDOC  000114    010  ERR3    000274R             FF.RWD=  000001          F.MBCT=  000054          N.FID  =  000100
B.HRLP  000126    010  ERR3L =  000041             FF.RWF=  000006          F.MBC1=  000055          N.FNAM=  000006
B.HRLR  000122    010  ERR3T   000276R             FF.SPC=  000004          F.MBFG=  000056          N.FOS  =  000764
B.HRLW  000124    010  ERR4    000340R             FILERR  002124R          F.NRBD=  000024          N.FTYP=  000014
B.NMBR  000052    010  ERR4L =  000022             FN.DBR  000026    011 F.NREC=  000030          N.FVER=  000016
```

```
DMCIN  MACRO M1110  27-MAR-80 13:24  PAGE 16-4
SYMBOL TABLE
```

```
N.NEXT= 000022          SR.ARS  000106      002 STRTCT  001214R          S.FNB = 000036          XBATCH= 000013
N.PKSZ= 000020          SR.DAY  000010      002 ST.ASZ  000020      006 S.FNBW= 000017          XDBLOA= 000004
N.PKTS= 000043          SR.DLT  000014      002 ST.BSZ  000024      006 S.FNTY= 000004          XDBPRO= 000012
N.QURY= 000031          SR.ECB  000047      002 ST.BTC  000000      006 S.FTYP= 000002          XDMCIN= 000006
N.STAT= 000020          SR.ECH  000046      002 ST.CSZ  000030      006 S.HRL = 000240          XFOSMR= 000007
N.SUNT= 000002          SR.ECL  000050      002 ST.HRL  000010      006 S.NFEN= 000020          XGTSRE= 000014
N.UNIT= 000034          SR.FIB  000012      002 ST.LEN  000044      006 TABLSZ= 000007          XHITSK= 000011
OFILE   000010R         SR.GRE  000100      002 ST.QRY  000002      006 TNXTBF  001140R        XHLMER= 000002
OPENFL  001760R         SR.GRS  000072      002 ST.QSZ  000034      006 TSKERR  001452R        XHOTSK= 000010
PAR$$$= 000027          SR.LEN  000122      002 ST.SCH  000040      006 TSKGD   001462R        XMLUN = 000002
PUTSSO= ****** G        SR.LIN  000066      002 ST.UHL  000004      006 UNITNB  000502R        XMSCHE= 000000
QE.ROI= 000144          SR.LIP  000062      002 ST.XLT  000014      006 WAITBF  001166R        XQTS  = 000003
Q.FDSC  000004      007 SR.MON  000006      002 SUDHRI= ****** G        WN.NTP  000004    012 XQT0  = 000001
Q.IOAE= 000012          SR.NDC  000042      002 SUINDX= ****** G        WN.NXT  000006    012 XSULOA= 000005
Q.IOEF= 000006          SR.NDS  000036      002 SUST  = ****** G        WN.ROT  000002    012 $CBDSG= ****** GX
Q.IOFN= 000002          SR.NIN  000030      002 SUXDON  000736R         WN.SIZ  000010    012 $DSW  = ****** GX
Q.IOLU= 000004          SR.NIP  000022      002 SU.DBU= 000004          WN.SRC  000000    012 $$$   = 000072R     015
Q.IOPL= 000014          SR.SDB  000032      002 SU.DON= 000006          WN.TYP  000001    012 $$$ARG= 000002
Q.IOPR= 000007          SR.SRC  000002      002 SU.IDL= 000000          WORD0 = 000000        $$$OST= 000006
Q.IOSB= 000010          SR.SUN  000000      002 SU.LOD= 000001          WORD1 = 000002        $$$T1 = 000005
Q.NQBK  000000      007 SR.TWS  000056      002 SU.SRC= 000002          WORD2 = 000004        .CLOSE= ****** G
Q.NUHL  000002      007 SR.WSL  000052      002 SU.SRR= 000005          WORD3 = 000006        .DLFNB= ****** G
Q.SIZE  000014      007 SR.YR   000004      002 SU.XPD= 000003          WORD4 = 000010        .FINIT= ****** G
READMC  001504R         SP.1IN  000024      002 SYIOS   000004R        WORD5 = 000012        .FSRCB= ****** G
R.FIX = 000001          SR.1IP  000016      002 S.BFHD= 000020          WORD6 = 000014        .OPFNB= ****** G
R.SEQ = 000003          SS.FID  000002      004 S.DABA= 000006          WORD7 = 000016        .WAIT = ****** G
R.SUTN= 000002          SS.FNB  000010      004 S.DAEF= 000010          WORD8 = 000020        .WRITE= ****** G
R.VAR = 000002          SS.FVR  000006      004 S.DATN= 000002          WORD9 = 000022        ...GBL= 000000
SDATA   000450R         SS.LEN  000012      004 S.FATT= 000016          WRDVAL= 000024        ...PC1= 000010R
SN    = ****** G        SS.STT  000000      004 S.FDB = 000140          WRTBLK  002014R       ...PC2= 000150R
SRECPT= ****** G        STATRC  000766R         S.FNAM= 000006          WRTFIL  001700R       ...TPC= 000020
SR.ARE  000114      002
```

```
. ABS.  000000      000
        002246      001
SRCOFF  000122      002
FDSCOF  000010      003
SUSOFF  000012      004
DHROFF  000012      005
STTOFF  000044      006
QSPLOF  000014      007
BSTOFF  000772      010
FNOFFS  000044      011
WNODOF  000010      012
DNODOF  000010      013
$$FSR1  000000      014
$DPB$$  000100      015
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  7482 WORDS  ( 30 PAGES)
DYNAMIC MEMORY:  9140 WORDS  ( 35 PAGES)
ELAPSED TIME:  00:00:57
DMCIN,DMCIN/-SP=[20,1]P,M,DMCIN
```

TABLE OF CONTENTS

10-   2   MACRO'S AND CONSTANTS
11-   21   ASSEMBLY-TIME DATA DEFINITIONS
12-   79   GET RECEIVE PACKET, INITIALIZE, SEND FILES

```
                                .TITLE  SULOAD
                                .SBTTL  MACRO'S AND CONSTANTS
                        ;
                                .MCALL  FDOF$L,FCSBT$,FDBDF$,FDRC$A
                                .MCALL  FDBK$A,FDOP$A,FSRSZ$,FINIT$
                                .MCALL  RCVX$C,ALUN$S,QIOW$C,QIOW$S,EXIT$S
                                .MCALL  SDAT$C,RQST$C
                                .MCALL  MRKT$C,WTSE$C
                                .MCALL  OFNB$R,READ$,WAIT$,CLOSE$
                        ;
                                .GLOBL  BLDEFL,BSTPTR
                        ;
                        ;LUNS
000003                  DPLUN=3
000006                  COLUN=6
                        ;
                        ;MISC EQUATES
000001                  EF.IO=1         ;I/O EVENT FLAG
                        ;
```

```
                                        .SBTTL  ASSEMBLY-TIME DATA DEFINITIONS
000000                                  .PSECT
                                ;
000000                                  FDOF$L
000000                                  FCSBT$
                                ;
                                ;INPUT FILE FDB
000000                          FDB:    FDBDF$
000140                                  FDRC$A  FD.RWM
000140                                  FDBK$A  DATBUF,N.BUFB,,EF,IO,IOSTAT
000140                                  FDOP$A  DPLUN
                                ;
000140                                  FSRSZ$  0
                                ;
                                ;MISC LOCATIONS
000140                          DATBUF: .BLKB   N.BUFB          ;BLOCK DATA BUFFER
004140                          IOSTAT: .BLKW   2               ;I/O STATUS BLOCK
004144  000001                  XMLUN:: .WORD   1               ;LUN 1 = XM0
                                                                ;LUN 2 = XM1
004146                          RDATA:  .BLKW   2               ;RECEIVE DATA FROM MSCHED
004152  000000                          .WORD   0               ;COMMAND
004154      000                         .BYTE   0               ;BATCH #
004155      000                         .BYTE   0               ;JUNK
004156  000000                          .WORD   0               ;SU#
004160                                  .BLKW   10.
004204      005     000         SDATA:  .BYTE   5,0             ;SEND DATA TO XSCHED
004206  000000                          .WORD   0
004210  000000                          .WORD   0
004212  000000                          .WORD   0
                                ;
                                ;
                                ;ERROR MESSAGES
004214  000022                  ERR1:   .WORD   ERR1L
004216      123     125     114 ERR1T:  .ASCII  /SULOAD: DISK ERROR/
    004221      117     101     104
    004224      072     040     104
    004227      111     123     113
    004232      040     105     122
    004235      122     117     122
          000022                ERR1L=.-ERR1T
                                        .EVEN
                                ;
004240  000031                  ERR2:   .WORD   ERR2L
004242      123     125     114 ERR2T:  .ASCII  /SULOAD: DMC WRITE FAILURE/
    004245      117     101     104
    004250      072     040     104
    004253      115     103     040
    004256      127     122     111
    004261      124     105     040
    004264      106     101     111
    004267      114     125     122
    004272      105
          000031                ERR2L=.-ERR2T
                                        .EVEN
                                ;
004274  000020                  ERR3:   .WORD   ERR3L
004276      123     125     114 ERR3T:  .ASCII  /SULOAD: LINK INL/
```

```
        004301      117     101     104
        004304      072     040     114
        004307      111     116     113
        004312      040     111     116
        004315      114
    65          000020                  ERR3L=.-ERR3T
    66                                  .EVEN
    67                                  ;
    68                                  ;
    69                                  ;
    70                                  ;BST FDSC OFFSET INDEX TABLE
    71                                  ;       .WORD   B.FMHR          ;SEND FIRST
    72 004316  000162                   FDSCID: .WORD   B.FQLS
    73 004320  000202                           .WORD   B.FFSA
    74 004322  000212                           .WORD   B.FFSB
    75 004324  000222                           .WORD   B.FFSC
    76 004326  000000                           .WORD   0               ;END OF TABLE
    77                                  ;
```

```
                                        .SBTTL  GET RECEIVE PACKET, INITIALIZE, SEND FILES
                                ;
004330                          SULOAD:
004330                                  FINIT$
                                ;
004334                          NXTRCV: RCVX$C  ,RDATA                  ;GET RECEIVE PACKET
                                ;
004342  016767  177610  177574          MOV     RDATA+8.,XMLUN
004350  062767  000001  177566          ADD     #1,XMLUN
                                ;
004356  116705  177572                  MOVB    RDATA+6,R5              ;BATCH NUMBER
004362  012704  004316'                 MOV     #FDSCID,R4              ;FDSC OFFSET TABLE
004366  016703  177564                  MOV     RDATA+8.,R3             ;R3 = SU NUMBER
                                ;
                                ;   FIRST, LOAD HRL TO SU
                                ;
004372                                  CALL    LDHRL                   ;LOAD HRL
                                ;
                                ;TOP OF LOOP FOR EACH CONTROL TABLE FILE
004376  012401                  NXTFIL: MOV     (R4)+,R1                ;FDSC OFFSET IN BST
004400  001534                          BEQ     LDONE                   ;ALL FIVE TABLES LOADED
004402  066501  000000G                 ADD     BSTPTR(R5),R1           ;FDSC ADDRESS IN BST
004406  012700  000000'                 MOV     #FDB,R0                 ;INPUT FILE FDB
                                ;
004412                                  CALL    BLDEFL                  ;BUILD FNB IN FDB
004416                                  OFNB$R                          ;OPEN FILE
004430  103433                          BCS     DSKERR                  ;OPEN FAILURE
004432  016002  000010                  MOV     F.EFBK(R0),R2           ;# OF SECTORS IN FILE
004436  016003  000012                  MOV     F.EFBK+2(R0),R3
004442  162703  000001                  SUB     #1,R3
004446  005602                          SBC     R2
004450  071227  000004                  DIV     #N.BFAC,R2              ;# OF BLOCKS
004454  001421                          BEQ     DSKERR                  ;EMPTY FILE
004456  005703                          TST     R3                      ;NOT MULTIPLE -
004460  001017                          BNE     DSKERR                  ;  OF N.BFAC
                                ;
                                ;TOP OF LOOP FOR EACH BLOCK
004462                                  READ$                           ;READ FIRST BLOCK
004466                                  WAIT$
004472  010267  173444                  MOV     R2,DATBUF+2             ;STUFF BLOCK COUNT IN FIRST BLOCK
004476  000404                          BR      TSTRED                  ;CONTINUE
                                ;
004500                          REDFIL: READ$                           ;READ NEXT BLOCK
004504                                  WAIT$
004510  122767  000000G 177422  TSTRED: CMPB    #IS.SUC,IOSTAT          ;SUCCESS$
004516  001403                          BEQ     DSKOK                   ;YES
004520  012700  004214'          DSKERR: MOV     #ERR1,R0                ;ERROR
004524  000514                          BR      ERROR
                                ;
004526                          DSKOK:  QIOW$S  #IO.WLB,@#XMLUN,#EF,IO,,#IOSTAT,,<#DATBUF,#N.BUFB>
004600  122767  000000G 177332          CMPB    #IS.SUC,IOSTAT          ;SUCCESS?
004606  001423                          BEQ     1$                      ;YES
004610  122767  000000G 177322          CMPB    #IE.CNR,IOSTAT          ;RECOVERABLE ERROR?
004616  001443                          BEQ     LNKRST
004620  122767  000000G 177312          CMPB    #IE.DNR,IOSTAT
004626  001437                          BEQ     LNKRST
004630  122767  000000G 177302          CMPB    #IE.TMO,IOSTAT
```

```
SULOAD  M[illegible]D.MI110  27-MAR-80 13:19  PAGE 12-1
GET RECEIVE PACKET, INITIALIZE, SEND FILES
```

```
     136 004636  001433                          BEQ     LNKRST
     137 004640  122767  000000G 177272          CMPB    #IE.ABO,IOSTAT
     138 004646  001427                          BEQ     LNKRST
     139 004650  012700  004240'                 MOV     #ERR2,R0                ;ERROR
     140 004654  000440                          BR      ERROR
     141                                 ;
     142 004656  077270                  1$:     SOB     R2,REDFIL               ;CONTINUE WITH FILE
     143                                 ;
     144                                 ;READ ENTIRE FILE
     145 004660                          CLSFIL: CLOSE$  #FDB                    ;CLOSE FILE
     146 004670  000642                          BR      NXTFIL                  ;GET NEXT CONTROL TABLE FILE
     147                                 ;
     148                                 ;
     149                                 ;       ACK SCHEDULER, ALL THE TABLES LOADED
     150                                 ;
     151 004672                          LDONE:
     152 004672                                  CALL    GETFRE                  ;GET FREE PACKET
     153 004676  012762  000005  000002          MOV     #5,2(R2)                ;PUT IN THE COMMAND SOURCE...
     154 004704  116762  177244  000004          MOVB    RDATA+6.,4(R2)          ;...BATCH NO...
     155 004712  016762  177240  000006          MOV     RDATA+8.,6(R2)          ;...AND SERACH UNIT NO.
     156 004720                                  CALL    PUTSSQ                  ;QUEUE IT TO MSCHED
     157 004724  000603                          BR      NXTRCV                  ;GO GET ANOTHER PACKET
     158                                 ;
     159                                 ;
     160                                 ;
     161                                 ;RECOVERABLE LINK ERROR
     162                                 ;
     163 004726  012700  004274'          LNKRST: MOV     #ERR3,R0                ;ERROR MESSAGE
     164 004732                                  CALL    COOUT
     165 004736                                  MRKT$C  EF,IO,60.,1             ;WAIT FOR OTHER END TO RECOVER
     166 004744                                  WTSE$C  EF,IO
     167 004752  005744                          TST     -(R4)                   ;BACK OFF TABLE POINTER
     168 004754  000741                          BR      CLSFIL                  ;CLOSE FILE AND RESEND
     169                                 ;
     170                                 ;
     171                                 ;ERROR MESSAGE EXIT - R0:ERROR MESSAGE ADDRESS
     172                                 ;
     173 004756                          ERROR:  CALL    COOUT                   ;MSG TO CONSOLE
     174 004762                          EXIT:   EXIT$S
     175                                 ;
     176                                 ;
     177                                 ;SEND MESSAGE TO CONSOLE
     178                                 ;
     179 004770  012001                  COOUT:  MOV     (R0)+,R1                ;MSG ADDRESS AND LENGTH
     180 004772                                  QIOW$S  #IO,WLB,#COLUN,#EF,IO,,,,<R0,R1,#40>    ;MSG TO CONSOLE
     181 005040  000207                          RTS     PC
     182                                 ;
     183                                 ;
     184         004330'                         .END    SULOAD
```

```
BITVAL=  000000          B.QMAP   000234      010 FA.RD  =  000001          FN.SF0   000030      011 IE.ABO=  ****** GX
BIT0   =  000001          B.QSPL   000316      010 FA.RWD=  004000          FN.SF1   000032      011 IE.CNR=  ****** GX
BIT1   =  000002          B.QTTM   000076      010 FA.SEQ=  040000          FN.SHD   000042      011 IE.DNR=  ****** GX
BIT10  =  002000          B.QUQP   000056      010 FA.SHR=  000040          FO.APD=  000106          IE.TMO=  ****** GX
BIT11  =  004000          B.SFDB   000010      010 FA.TMP=  000020          FO.MFY=  000002          IOSTAT   004140R
BIT12  =  010000          B.SIZE   000772      010 FA.WCK=  020000          FO.RD  =  000001          IO.WLB=  ****** GX
BIT13  =  020000          B.SNDP   000012      010 FA.WRT=  000002          FO.UPD=  000006          IS.SUC=  ****** GX
BIT14  =  040000          B.SSQ    000004      010 FDB    =     000000R     FO.WRT=  000016          LDHRL  =  ****** GX
BIT15  =  100000          B.SSQF   000050      010 FDSCID   004316R         F.ACTL=  000076          LDONE    004672R
BIT2   =  000004          B.STAT   000044      010 FD.BLK=  000010          F.ALOC=  000040          LNKRST   004726R
BIT3   =  000010          B.STTE   000053      010 FD.CCL=  000002          F.BBFS=  000062          M      =  000062
BIT4   =  000020          B.UDOC   000110      010 FD.COM=  020000          F.BDB  =  000070          M.KTAE=  000010
BIT5   =  000040          CF.B0  =  000070          FD.CR  =  000002          F.BGBC=  000057          M.KTEF=  000002
BIT6   =  000100          CF.B2  =  000067          FD.DIR=  000010          F.BKDN=  000026          M.KTMG=  000004
BIT7   =  000200          CF.B4  =  000066          FD.FID   000000      003 F.BKDS=  000020          M.KTUN=  000006
BIT8   =  000400          CF.B6  =  000065          FD.FNB   000006      003 F.BKEF=  000050          N      =  000002
BIT9   =  001000          CF.DR0=  000064          FD.FTN=  000001          F.BKP1=  000051          NB.DEV=  000200
BLDEFL=  ****** G         CF.DR1=  000063          FD.FVR   000004      003 F.BKST=  000024          NB.DIR=  000100
BSTPTR=  ****** G         CH.AND=  000001          FD.F11=  040000          F.BKVB=  000064          NB.NAM=  000004
BS.CLS=  000002           CLSFIL   004660R         FD.INS=  000010          F.CHR  =  000075          NB.SD1=  000400
BS.DBU=  000004           COLUN  =  000006          FD.ISP=  002000          F.CNTG=  000034          NB.SD2=  001000
BS.INA=  000000           COOUT    004770R         FD.LEN   000010      003 F.DFNB=  000046          NB.SNM=  000040
BS.OPN=  000001           DATBUF   000140R         FD.MNT=  100000          F.DSPT=  000044          NB.STP=  000020
BS.SRC=  000003           DBGLEN=  000116          FD.OSP=  004000          F.DVNM=  000134          NB.SVR=  000010
BYTE0  =  000000          DH.BF0   000002          FD.PLC=  000004          F.EFBK=  000010          NB.TYP=  000002
BYTE1  =  000001          DH.BF1   000004          FD.PRN=  000004          F.EFN  =  000050          NB.VER=  000001
BYTE2  =  000002          DH.CTL   000000          FD.PSE=  010000          F.EOBB=  000032          NXTFIL   004376R
BYTE3  =  000003          DH.DMC   000010          FD.RAH=  000001          F.ERR  =  000052          NXTRCV   004334R
BYTE4  =  000004          DH.FLG   000006          FD.RAN=  000002          F.FACC=  000043          N.BFAC=  000004
BYTE5  =  000005          DN.DCK   000000          FD.REC=  000001          F.FFBY=  000014          N.BHGH=  000006
BYTE6  =  000006          DN.NTP   000004          FD.RWM=  000001          F.FNAM=  000110          N.BTCH=  000004
BYTE7  =  000007          DN.NXT   000006          FD.SDI=  000020          F.FNB  =  000102          N.BUFB=  004000
BYTE8  =  000010          DN.ROT   000002          FD.SQD=  000040          F.FTYP=  000116          N.BUFW=  002000
BYTE9  =  000011          DN.SIZ   000010          FD.TTY=  000004          F.FVER=  000120          N.DID  =  000024
BYTVAL=  000012           DPLUN  =  000003          FD.WBH=  000002          F.HIBK=  000004          N.DVNM=  000032
B.BSTA   000054      010 DSKERR   004520R         FF.CHR=  000005          F.LUN  =  000042          N.FID  =  000000
B.CNTX   000046      010 DSKOK    004526R         FF.NV  =  000003          F.MBCT=  000054          N.FNAM=  000006
B.CQUQ   000060      010 EF.IO  =  000001          FF.POE=  000002          F.MBC1=  000055          N.FOS  =  000764
B.FEMA   000132      010 ERROR    004756R         FF.RWD=  000001          F.MBFG=  000056          N.FTYP=  000014
B.FEMB   000142      010 ERR1     004214R         FF.RWF=  000006          F.NRBD=  000024          N.FVER=  000016
B.FEMC   000152      010 ERR1L  =  000022          FF.SPC=  000004          F.NREC=  000030          N.NEXT=  000022
B.FFSA   000202      010 ERR1T    004216R         FN.DBR   000026      011 F.OVBS=  000030          N.PKSZ=  000020
B.FFSB   000212      010 ERR2     004240R         FN.DBS   000022      011 F.RACC=  000016          N.PKTS=  000043
B.FFSC   000222      010 ERR2L  =  000031          FN.DHR   000040      011 F.RATT=  000001          N.QURY=  000031
B.FMHR   000172      010 ERR2T    004242R         FN.EMA   000012      011 F.RCNM=  000034          N.STAT=  000020
B.FQLS   000162      010 ERR3     004274R         FN.EMB   000014      011 F.RCTL=  000017          N.SUNI=  000002
B.FSAZ   000100      010 ERR3L  =  000020          FN.EMC   000016      011 F.RSIZ=  000002          N.UNIT=  000034
B.FSBZ   000102      010 ERR3T    004276R         FN.FSA   000000      011 F.RTYP=  000000          PAR$$$=  000027
B.FSCZ   000104      010 EXIT     004762R         FN.FSB   000002      011 F.SEQN=  000100          PUTSSQ=  ****** GX
B.HBLK   000120      010 FA.APD=  000100          FN.FSC   000004      011 F.SPDV=  000072          QE.RO1=  000144
B.HDOC   000114      010 FA.CRE=  000010          FN.LGO   000034      011 F.SPUN=  000074          Q.FDSC   000004      007
B.HRLP   000126      010 FA.DLK=  001000          FN.LGU   000036      011 F.STBK=  000036          Q.NQBK   000000      007
B.HRLR   000122      010 FA.ENB=  100000          FN.MFO   000024      011 F.UNIT=  000136          Q.NUHL   000002      007
B.HRLW   000124      010 FA.EXC=  002000          FN.MHR   000010      011 F.URBD=  000020          Q.SIZE   000014      007
B.NMBR   000052      010 FA.EXT=  000004          FN.NMB   000044      011 F.VBN  =  000064          RDMMA    004146R
B.NQRY   000232      010 FA.NSP=  000100          FN.QLS   000006      011 F.VBSZ=  000060          REDFILE  004580R
B.QLSZ   000106      010 FA.POS=  010000          FN.QRY   000020      011 GETFRE=  ****** GX          R.FIX  =  000001
```

```
SULOAD  M[illegible]0.M1110  27-MAR-80 13:19  PAGE 12-3
SYMBOL TA[illegible]
```

```
R.SEQ = 000003          SR.SDB  000032      002 ST.XLT  000014      006 WN.ROT  000002      012 XHITSK = 000011
R.VAR = 000002          SR.SRC  000002      002 SULOAD  004330R         WN.SIZ  000010      012 XHLMER = 000002
R.VXBA = 000006         SR.SUN  000000      002 SU.DBU = 000004         WN.SRC  000000      012 XHOTSK = 000010
R.VXTN = 000002         SR.TWS  000056      002 SU.DON = 000006         WN.TYP  000001      012 XMLUN   004144RG
SDATA   004204R         SR.WSL  000052      002 SU.IDL = 000000         WORD0 = 000000          XMSCHE = 000000
SR.ARE  000114      002 SR.YR   000004      002 SU.LOD = 000001         WORD1 = 000002          XQTS   = 000003
SR.ARS  000106      002 SR.1IN  000024      002 SU.SRC = 000002         WORD2 = 000004          XQT0   = 000001
SR.DAY  000010      002 SR.1IP  000016      002 SU.SRR = 000005         WORD3 = 000006          XSULOA = 000005
SR.DLT  000014      002 SS.FID  000002      004 SU.XPD = 000003         WORD4 = 000010          $$$    = 000022R    015
SR.ECB  000047      002 SS.FNB  000010      004 S.BFHD = 000020         WORD5 = 000012          $$$ARG = 000002
SR.ECH  000046      002 SS.FVR  000006      004 S.FATT = 000016         WORD6 = 000014          $$$OST = 000004
SR.ECL  000050      002 SS.LEN  000012      004 S.FDB = 000140          WORD7 = 000016          $$$T1  = 000000
SR.FIB  000012      002 SS.STT  000000      004 S.FNAM = 000006         WORD8 = 000020          .CLOSE = ****** G
SR.GRE  000100      002 ST.ASZ  000020      006 S.FNB = 000036          WORD9 = 000022          .FINIT = ****** G
SR.GRS  000072      002 ST.BSZ  000024      006 S.FNBW = 000017         WRDVAL = 000024         .FSRCB = ****** G
SR.LEN  000122      002 ST.BTC  000000      006 S.FNTY = 000004         W.TSEF = 000002         .OPFNB = ****** G
SR.LIN  000066      002 ST.CSZ  000030      006 S.FTYP = 000002         XBATCH = 000013         .READ  = ****** G
SR.LIP  000062      002 ST.HRL  000010      006 S.HRL = 000240          XDBLOA = 000004         .WAIT  = ****** G
SR.MON  000006      002 ST.LEN  000044      006 S.NFEN = 000020         XDBPRO = 000012             ...GBL = 000000
SR.NDC  000042      002 ST.ORY  000002      006 TSTRED  004510R         XDMCIN = 000006             ...PC1 = 000000R
SR.NDS  000036      002 ST.OSZ  000034      006 WN.NTP  000004      012 XFOSMR = 000007             ...PC2 = 000140R
SR.NIN  000030      002 ST.SCH  000040      006 WN.NXT  000006      012 XGTSRE = 000014             ...TPC = 000020
SR.NIP  000022      002 ST.UHL  000004      006
```

```
. ABS.  000000     000
        005042     001
SRCOFF  000122     002
FDSCOF  000010     003
SUSOFF  000012     004
DHROFF  000012     005
STTOFF  000044     006
QSPLOF  000014     007
BSTOFF  000772     010
FNOFFS  000044     011
WNODOF  000010     012
DNODOF  000010     013
$$FSR1  000000     014
$DPB$$  000026     015
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  7159 WORDS  (28 PAGES)
DYNAMIC MEMORY:  8084 WORDS  ( 31 PAGES)
ELAPSED TIME:  00:00:47
SULOAD,SULOAD/-SP=[20,1]DP,M,SULOAD
```

11- 34 ASSEMBLY-TIME·DATA·DEFINITIONS·
15- 184 BUFFER·CONTROL ROUTINES·

```
                                        .TITLE  LDHRL
                                ;
                                        .MCALL  FDOF$L,FCSBT$,FDBDF$,FDRC$A,FDBK$R
                                        .MCALL  FDBK$A,FDOP$A,FSRSZ$,FINIT$
                                        .MCALL  RCVX$C,ALUN$S,QIOW$,QIOW$S,EXIT$S
                                        .MCALL  SDAT$C,RQST$C,DIR$,ASTX$S
                                        .MCALL  MRKT$C,WTSE$C
                                        .MCALL  OFNB$R,READ$,WAIT$,CLOSE$,WRITE$
                                ;
                                        .GLOBL  BLDEFL,BSTPTR
                                ;
                                ;       LUNS
000003                          DPLUN=3
000006                          COLUN=6
                                ;
                                ;       MISC EQUATES
000001                          EF,IO=1         ;I/O EVENT FLAG
                                ;
000000                                  .PSECT  IOCBOF,ABS
000000                          IOST:   .BLKW   2               ;I/O STATUS BLOCK
000004                          FDBAD:  .BLKW   1               ;ADDRESS OF FDB
000006                          EVNT:   .BLKB   1               ;ADDRESS OF EVENT FLAG
000007                          ATTR:   .BLKB   1               ;TYPE OF I/O
000010                          CBLK:   .BLKW   1               ;CURRENT LOGICAL BLOCK
000012                          LBLK:   .BLKW   1               ;LAST BLOCK (IF WRAP AROUND)
000014                          CBUF:   .BLKW   1               ;ADDRESS OF BUFFER CURRENTLY IN USE
000016                          NXTG:   .BLKW   1               ;BDB OFFSET OF NEXT ENTRY FOR ASSIGNMENT
000020                          NXTR:   .BLKW   1               ;BDB OFFSET OF NEXT ENTRY FOR I/O
000022                          IBDB:   .BLKB   1               ;FILE'S FIRST BDB INDEX
000023                          LBDB:   .BLKB   1               ;FILE'S LAST BDB INDEX
000000                                  .PSECT
                                ;
```

```
                                        .SBTTL  ASSEMBLY-TIME DATA DEFINITIONS
000000                                  .PSECT
                                ;
000000                                  FDOF$L
000000                                  FCSBT$
                                ;
                                ;       INPUT FILE FDB
000000                          FDB:    FDBDF$
000140                                  FDRC$A  FD,RWM
000140                                  FDBK$A  ,N,BUFB,,EF,IO,IOCB
000140                                  FDOP$A  DPLUN
                                ;
000140                                  FSRSZ$  0
                                ;
                                ;       I/O CONTROL BLOCK - HRL FILE INPUT
                                ;
000140                          IOCB:
000140                                  .BLKW   2                       ;IOST BLOCK
000144  000000'                         .WORD   FDB                     ;FDB ADDRESS
000146      001                         .BYTE   EF,IO                   ;EVENT FLAG
000147      024                         .BYTE   BUFRD!BUFRAP            ;READ WITH WRAP AROUND
000150                                  .BLKW   1                       ;CURRENT LOGICAL BLOCK
000152                                  .BLKW   1                       ;LAST LOGICAL BLOCK
000154  000000                          .WORD   0                       ;ADDRESS OF BUFFER IN USE
000156  000000                          .WORD   BDB1-BDB                ;INDEX TO INITIAL ASSIGN BDB
000160  000000                          .WORD   BDB1-BDB                ;INDEX TO INITIAL I/O BDB
000162      000                         .BYTE   BDB1-BDB                ;INDEX TO FILE'S FIRST BDB
000163      002                         .BYTE   BDB2-BDB                ;INDEX TO FILE'S LAST BDB
                                ;
                                ;       BUFFER DEFINITION BLOCK ADDRESS VECTOR
000164                          BDB:
000164  000170'                 BDB1:   .WORD   BUF1
000166  004170'                 BDB2:   .WORD   BUF2
                                ;
                                ;       I/O BUFFERS
                                ;
000170                          BUF1:   .BLKW   N.BUFW
004170                          BUF2:   .BLKW   N.BUFW
                                ;
                                ;       MISC LOCATIONS
010170  000000                  BSTADR: .WORD   0
010172  000000                  R1SAV:  .WORD   0
                                ;
010174                          IOSTAT: .BLKW   2
```

```
                                        ;       ERROR.MESSAGES.
010200  000021                          ERR1:   .WORD.  ERR1L.
010202.     114     104     110         ERR1T:  .ASCII. /LDHRL: DISK.ERROR/.
010205      122.    114     072.
010210      040     104     111
010213      123     113     040
010216      105     122     122.
010221      117     122
            000021                      ERR1L=.-ERR1T.
                                                .EVEN.
                                        ;
010224  000030                          ERR2:   .WORD.  ERR2L.
010226      114     104     110         ERR2T:  .ASCII. /LDHRL: DMC.WRITE.FAILURE/.
010231      122.    114     072.
010234      040     104     115
010237      103     040     127
010242.     122.    111     124
010245      105     040     106
010250      101     111     114
010253      125     122     105
            000030                      ERR2L=.-ERR2T.
                                                .EVEN.
                                        ;
010256  000017                          ERR3:   .WORD.  ERR3L.
010260      114     104     110         ERR3T:  .ASCII. /LDHRL: LINK.INL/.
010263      122.    114     072.
010266      040     114     111
010271      116     113     040
010274      111     116     114
            000017                      ERR3L=.-ERR3T.
                                                .EVEN.
                                        ;
```

```
     96 010300                                  _DHRL::
     97 010300                                          SAVE    R0,R1,R2,R3,R4,R5
     98 010314  016502  000000G                         MOV     BSTPTR(R5),R2           ;GET BATCH STATUS TABLE
     99 010320  005762  000122                          TST     B.HRLR(R2)              ;IS THERE AN HRL FILE?
    100 010324  001002                                  BNE     10$                     ;BRANCH IF SO
    101 010326  000167  000430                          JMP     LDXIT                   ;EXIT IF NOT
    102 010332                                  10$:
    103 010332  010267  177632                          MOV     R2,BSTADR               ;SAVE BST ADDRESS
    104 010336  012701  000172                          MOV     #B.FMHR,R1              ;GET HRL'S FDSC ADDRESS
    105 010342  060201                                  ADD     R2,R1
    106 010344  012700  000000'                         MOV     #FDB,R0                 ;INPUT FILE FDB
    107 010350                                          CALL    BLDEFL                  ;BUILD FNB IN FDB
    108 010354                                          OFNB$R                          ;OPEN FILE
    109 010366  103436                                  BCS     DSKERR                  ;BRANCH IF OPEN FAILUR
    110                                         ;
    111 010370  012701  000140'                         MOV     #IOCB,R1                ;LOAD I/O CONTROL BLOCK
    112 010374                                          CALL    GTBLK                   ;GET ALLOCATION SIZE OF HRL FILE
    113 010400  166203  000120                          SUB     B.HBLK(R2),R3           ;CALCULATE USED BLOCKS
    114 010404  003510                                  BLE     CLSFIL                  ;TRIVIAL HRL FILE - SUPPRESS LOAD
    115 010406  016261  000122  000010                  MOV     B.HRLR(R2),CBLK(R1)     ;LOAD START BLOCK
    116 010414  016261  000124  000012                  MOV     B.HRLW(R2),LBLK(R1)     ;LOAD STOP BLOCK
    117 010422                                          CALL    BUFIO                   ;PRIME INPUT BUFFERS
    118                                         ;
    119                                         ;       TOP OF LOOP FOR EACH BLOCK
    120                                         ;
    121 010426                                  TOPLUP:
    122 010426                                          CALL    GETBUF                  ;REQUEST INPUT BUFFER
    123 010432  016104  000014                          MOV     CBUF(R1),R4             ;GET ADDRESS OF BUFFER
    124 010436  010364  000002                          MOV     R3,2(R4)                ;STUFF BLOCK COUNT IN FIRST BLOCK
    125 010442  000404                                  BR      TSTRED                  ;CONTINUE
    126                                         ;
    127 010444                                  REDFIL:
    128 010444                                          CALL    GETBUF                  ;REQUEST INPUT BUFFER
    129 010450  016104  000014                          MOV     CBUF(R1),R4             ;GET ADDRESS OF BUFFER
    130 010454  122767  000000G 167456  TSTRED: CMPB    #IS.SUC,IOCB            ;SUCCESS$
    131 010462  001403                                  BEQ     DSKOK                   ;YES
    132 010464  012700  010200'         DSKERR: MOV     #ERR1,R0                ;ERROR
    133 010470  000502                                  BR      ERROR
    134                                         ;
    135 010472                                  DSKOK:
    136 010472                                          QIOW$S  #IO.WLB,0#XMLUN,#EF,IO,,#IOSTAT,,<R4,#N,BUFB>
    137 010542  122767  000000G 177424                   CMPB    #IS.SUC,IOSTAT          ;SUCCESS?
    138 010550  001423                                  BEQ     1$                      ;YES
    139 010552  122767  000000G 177414                   CMPB    #IE.CNR,IOSTAT          ;RECOVERABLE ERROR?
    140 010560  001427                                  BEQ     LNKRST
    141 010562  122767  000000G 177404                   CMPB    #IE.DNR,IOSTAT
    142 010570  001423                                  BEQ     LNKRST
    143 010572  122767  00H000G 177374                   CMPB    #IE.TMO,IOSTAT
    144 010600  001417                                  BEQ     LNKRST
    145 010602  122767  000000G 177364                   CMPB    #IE.ABO,IOSTAT
    146 010610  001413                                  BEQ     LNKRST
    147 010612  012700  010224'                         MOV     #ERR2,R0                ;ERROR
    148 010616  000427                                  BR      ERROR
    149                                         ;
    150 010620                                  1$:
    151 010620                                          CALL    RELBUF                  ;RELEASE BUFFER
    152 010624  077371                                  SOB     R3,REDFIL               ;CONTINUE WITH FILE
```

```
                                        ;       HAVE READ ENTIRE FILE
                                        ;
010626                                  CLSFIL: CLOSE$  #FDB                    ;CLOSE FILE
010636  000451                                 BR      LDXIT
                                        ;
                                        ;       RECOVERABLE LINK ERROR
                                        ;
010640  012700  010256'                 LNKRST: MOV     #ERR3,R0                ;ERROR MESSAGE
010644                                          CALL    COOUT
010650                                          MRKT$C  EF,IO,60.,1             ;WAIT FOR OTHER END TO RECOVER
010656                                          WTSE$C  EF,IO
010664                                          CLOSE$  #FDB
010674  000654                                  BR      TOPLUP                  ;CLOSE FILE AND RESEND
                                        ;
                                        ;       ERROR MESSAGE EXIT - R0:ERROR MESSAGE ADDRESS
                                        ;
010676                                  ERROR:  CALL    COOUT                   ;MSG TO CONSOLE
010702                                  EXIT:   EXIT$S
                                        ;
                                        ;       SEND MESSAGE TO CONSOLE
                                        ;
010710  012001                          COOUT:  MOV     (R0)+,R1                ;MSG ADDRESS AND LENGTH
010712                                          QIOW$S  #IO,WLB,#COLUN,#EF,IO,,,,<R0,R1,#40>    ;MSG TO CONSOLE
010760  000207                                  RETURN
                                        ;
010762                                  LDXIT:
010762  000241                                  CLC
010764                                          RESTOR  R0,R1,R2,R3,R4,R5
011000  000207                                  RETURN
```

```
                                                .SBTTL  BUFFER CONTROL ROUTINES
                                        ;
                                        ;       GET/RELEASE BUFFER
                                        ;       THESE SUBROUTINES ARE REENTRANT AND REQUIRE THAT R1 -> DESIRED IOCB
                                        ;
011002                                  GETBUF:
011002                                          SAVE    R2
011004  116102  000016                          MOVB    NXTG(R1),R2             ;GET OFFSET TO INTENDED BUFFER
011010  032762  000001  000164'                 BIT     #BUFLOK,BDB(R2)         ;IS NEXTG BUFFER ASSIGNABLE YET
011016                                          BOFF    20$                     ;BRANCH IF IT IS
011020  105761  000021                          TSTB    NXTR+1(R1)              ;HAS I/O BEEN STARTED ON THIS IOCB?
011024  001002                                  BNE     10$                     ;IF IT HAS, WE ARE I/O BOUND - WAIT
011026                                          CALL    BUFIO                   ;IF NOT, INPUT BUFFERS NEED TO BE PRIMED
011032                                  10$:
011032                                          WAIT$   FDBAD(R1),EVNT(R1)      ;WAIT FOR I/O COMPLETION
011050                                  20$:
                                        ;
011050  016261  000164' 000014                  MOV     BDB(R2),CBUF(R1)        ;LOAD ADDRESS OF ASSIGNED BUFFER
011056  000423                                  BR      BUFXIT
                                        ;
011060                                  RELBUF:
011060                                          SAVE    R2
011062  116102  000016                          MOVB    NXTG(R1),R2             ;GET OFFSET TO INTENDED BUFFER
011066  052762  000001  000164'                 BIS     #BUFLOK,BDB(R2)         ;MARK BUFFER UNASSIGNABLE
011074                                          CALL    BUFIO                   ;START UP I/O IF POSSIBLE
011100  126161  000016  000023                  CMPB    NXTG(R1),LBDB(R1)       ;LAST BUFFER FOR THIS FILE?
011106  002004                                  BGE     15$                     ;RECYCLE IF YES
011110  062761  000002  000016                  ADD     #2,NXTG(R1)             ;SELECT NEXT BUFFER TO BE ASSIGNED
011116  000403                                  BR      BUFXIT
011120                                  15$:
011120  116161  000022  000016                  MOVB    IBDB(R1),NXTG(R1)       ;SELECT FIRST BUFFER
011126                                  BUFXIT:
011126                                          RESTOR  R2
011130  000207                                  RETURN
```

```
                                        ;
                                        ;       ASYNCHRONOUS EXIT ROUTINE
                                        ;
011132                                  ASTX:
011132  010167  177034                          MOV     R1,R1SAV                ;SAVE R1
011136  011601                          MOV     (SP),R1                 ;GET IOST (SAME AS IOCB) ADDRESS FROM STACK
011140  116101  000020                  MOVB    NXTR(R1),R1             ;GET OFFSET TO I/O ACTIVE BUFFER
011144  042761  000001  000164'         BIC     #BUFLOK,BDB(R1)         ;UNLOCK THE BUFFER
011152  012601                          MOV     (SP)+,R1                ;POP IOCB ADDRESS FROM STACK
011154  005261  000010                  INC     CBLK(R1)                ;POINT TO NEXT LOGICAL BLOCK
011160  105061  000021                  CLRB    NXTR+1(R1)              ;RESET I/O IN PROGRESS FLAG
011164  126161  000020  000023          CMPB    NXTR(R1),LBDB(R1)       ;WAS THIS THE LAST BUFFER FOR FILE?
011172  002004                          BGE     10$                     ;RECYCLE IF YES
011174  062761  000002  000020          ADD     #2,NXTR(R1)             ;SELECT NEXT BUFFER FOR I/O
011202  000403                          BR      20$
011204                                  10$:
011204  116161  000022  000020          MOVB    IBDB(R1),NXTR(R1)       ;NEXT I/O BUFFER IS FIRST
                                        ;
                                        ;       START I/O ON NEXT BUFFER
011212                                  20$:
011212                                  SAVE    R0
011214  016100  000004                  MOV     FDBAD(R1),R0            ;GET FDB OF COMPLETED I/O
011220                                  SAVE    F,BKDS+2(R0),F.BKST(R0) ;PRESERVE ASTX ROUTINE REENTRANCY
011230                                  SAVE    F,BKVB+2(R0),F.EFN(R0)
011240                                  CALL    BUFIO
011244                                  RESTOR  F,BKVB+2(R0),F.EFN(R0)
011254                                  RESTOR  F,BKDS+2(R0),F.BKST(R0)
011264                                  RESTOR  R0
011266  016701  176700                  MOV     R1SAV,R1
011272                                  ASTX$S
                                        ;
                                        ;
                                        ;       CALCULATE LAST PHYSICAL BLOCK OF FILE
                                        ;
011300                                  GTBLK:
011300  016100  000004                  MOV     FDBAD(R1),R0            ;GET FILE'S FDB
011304  016003  000006                  MOV     F,HIBK+2(R0),R3         ;GET LAST ALLOCATED BLOCK
011310  006203                          ASR     R3                      ;CONVERT TO LOGICAL BLOCKS
011312  006203                          ASR     R3                      ;BY DIVIDING BY 4
011314  000207                          RETURN
                                        ;
                                        ;
                                        ;       BUFFER CONTROL ROUTINE SYMBOLIC VARIABLES
                                        ;
000001                                  BUFLOK  =       BIT0            ;BUFFER LOCKED, CANNOT BE ASSIGNED
000004                                  BUFRD   =       BIT2            ;READ OPERATIONS ON THIS IOCB
000010                                  BUFWR   =       BIT3            ;WRITE OPERATIONS ON THIS IOCB
000020                                  BUFRAP  =       BIT4            ;FILE WRAP AROUND IN EFFECT
000040                                  BUFEOF  =       BIT5            ;LAST LOGICAL I/O BLOCK ENCOUNTERED
000200                                  BUFOPN  =       BIT7            ;IOCB HAS BEEN INITIALIZED
```

```
                                        ;       PERFORM I/O ON NEXT AVAILABLE BUFFER
                                        ;
                                        ;       R1 -> BUFFER DEFINITION BLOCK
                                        ;
                                        BUFIO:
                                                SAVE    R0,R2,R3,R4
                                                BITB    #BUFOPN,ATTR(R1)        ;HAS IOCB BEEN INITIALIZED YET?
                                                BEQ     BUFQIO                  ;IF SO, SET UP I/O OPERATION
                                                TST     CBLK(R1)                ;HAS LOGICAL START OF FILE BEEN SPECIFIED?
                                                BGT     10$                     ;BRANCH IF IT HAS
                                                MOV     #1,CBLK(R1)             ;SET START AT BEGINNING OF FILE
                                        10$:
                                                TST     LBLK(R1)                ;HAS LOGICAL END BEEN SPECIFIED?
                                                BGT     20$                     ;BRANCH IF IT HAS
                                                BITB    #BUFRAP,ATTR(R1)        ;HAS WRAP AROUND BEEN SPECIFIED?
                                                BNE     20$                     ;BRANCH IF IT HAS
                                                CALL    GTBLK                   ;CALCULATE LOGICAL BLOCK SIZE OF FILE
                                                MOV     R3,LBLK(R1)             ;SET UP LOGICAL BLOCK TO STOP I/O
                                        20$:
                                                BITB    #BUFRD,ATTR(R1)         ;IS THIS IOCB FOR READ OPERATIONS?
                                                BEQF    30$                     ;BRANCH IF NOT
                                                MOVB    IBDB(R1),R2             ;GET OFFSET TO FIRST BUFFER
                                        25$:
                                                INC     BDB(R2)                 ;MARK BUFFER UNASSIGNABLE
                                                ADD     #2,R2                   ;STEP TO NEXT BUFFER
                                                CMPB    R2,LBDB(R1)             ;LAST BUFFER BEEN MARKED?
                                                BLE     25$                     ;IF NOT, GO TO MARK IT
                                        30$:
                                                FDBK$R  FDBAD(R1),,,,,,#ASTX    ;DEFINE AST EXIT
                                                BISB    #BUFOPN,ATTR(R1)        ;SET IOCB INITIALIZED FLAG
                                        ;
                                        ;       SET UP I/O OPERATION
                                        ;
                                        BUFQIO:
                                                BITB    #BUFEOF,ATTR(R1)        ;HAS FILE'S LOGICAL EOF ALREADY BEEN REACHED?
                                                BEQ     BUFIOX                  ;IF SO, EXIT
                                                TSTB    NXTR+1(R1)              ;IS I/O ALREADY IN PROGRESS ON THIS IOCB?
                                                BNE     BUFIOX                  ;EXIT IF SO
                                                MOV     NXTR(R1),R2             ;GET OFFSET TO NEXT BUFFER FOR I/O
                                                BIT     #BUFLOK,BDB(R2)         ;IS BUFFER AVAILABLE FOR I/O?
                                                BOFF    BUFIOX                  ;BRANCH IF NOT
                                                MOV     BDB(R2),R3              ;GET BUFFER ADDRESS
                                                BIC     #BUFLOK,R3              ;CLEAR LOCK FLAG
                                                FDBK$R  FDBAD(R1),R3            ;DEFINE BUFFER IN THE FDB
                                                FDBK$R  ,,,,EVNT(R1),R1         ;DEFINE EVENT FLAG AND IOST BLOCK
```

```
                                        ;       DO THE APPROPRIATE I/O OPERATION
                                        ;
011536                                  10$:
011536  016103  000012                          MOV     LBLK(R1),R3             ;GET LAST LOGICAL BLOCK
011542  105261  000017                          INCB    NXTG+1(R1)              ;SET LOGICAL BLOCK INDICATOR
011546                                  20$:
011546  026103  000010                          CMP     CBLK(R1),R3             ;COMPARE CURRENT WITH LAST
011552  002427                                  BLT     50$                     ;DO I/O - NOT LAST BLOCK
011554  001420                                  BEQ     40$                     ;DO I/O - TEST FOR LAST BLOCK
                                        ;
011556  132761  000020  000007                  BITB    #BUFRAP,ATTR(R1)        ;LAST HIGH - FILE WRAP AROUND?
011564                                          BOFF    BUFIOX                  ;HAVE ALREADY REACHED END OF FILE
011566  105761  000017                          TSTB    NXTG+1(R1)              ;LOGICAL BLOCK INDICATOR SET?
011572  001004                                  BNE     30$                     ;BRANCH IF YES
011574  012761  000001  000010                  MOV     #1,CBLK(R1)             ;WRAP AROUND TO START OF FILE
011602  000755                                  BR      10$                     ;RETEST FOR LOGICAL EOF
011604                                  30$:
011604                                          CALL    GTBLK                   ;CALCULATE LAST PHYSICAL BLOCK IN FILE
011610  105061  000017                          CLRB    NXTG+1(R1)              ;RESET LOGICAL BLOCK INDICATOR
011614  000754                                  BR      20$                     ;RETEST FOR PHYSICAL END OF FILE
011616                                  40$:
011616  105761  000017                          TSTB    NXTG+1(R1)              ;WHICH END OF FILE IS BEING TESTED?
011622  001405                                  BEQ     60$                     ;BRANCH IF PHYSICAL EOF
011624  152761  000040  000007                  BISB    #BUFEOF,ATTR(R1)        ;SET LOGICAL EOF REACHED INDICATOR
011632                                  50$:
011632  105061  000017                          CLRB    NXTG+1(R1)              ;RESET LOGICAL BLOCK INDICATOR
                                        ;
                                        ;       CALCULATE CURRENT VIRTUAL BLOCK FROM CBLK FIELD
                                        ;
011636                                  60$:
011636  016103  000010                          MOV     CBLK(R1),R3             ;GET CURRENT LOGICAL BLOCK
011642  005303                                  DEC     R3
011644  006303                                  ASL     R3
011646  006303                                  ASL     R3
011650  005203                                  INC     R3                      ;VB = (CBLK-1)*4+1
011652  016100  000004                          MOV     FDBAD(R1),R0            ;GET FDB ADDRESS
011656  010360  000066                          MOV     R3,F.BKVB+2(R0)         ;INITIALIZE VIRTUAL BLOCK ADDRESS IN FDB
011662  105261  000021                          INCB    NXTR+1(R1)              ;SET I/O IN PROGRESS
011666  132761  000004  000007                  BITB    #BUFRD,ATTR(R1)         ;READ SPECIFIED?
011674                                          BOFF    70$                     ;BRANCH IF NOT
011676                                          READ$                           ;PERFORM THE READ
011702  000406                                  BR      80$                     ;CHECK SUCCESS
011704                                  70$:
011704  132761  000010  000007                  BITB    #BUFWR,ATTR(R1)         ;WAS WRITE SPECIFIED?
011712                                          BOFF    BUFIOX                  ;BRANCH IF NOT
011714                                          WRITE$                          ;PERFORM THE WRITE
011720                                  80$:
011720  103024                                  BCC     BUFIOX                  ;EXIT IF NO ERROR
```

```
                                        ;       ERROR·EXIT·ON·FCS·DIRECTIVE·
                                        ;
                                        ERRX:
011722·  126027  000052  000000G·               CMPB·   F,ERR(R0),#IE.EOF·      ;WAS·THIS·READ·PAST·LAST·BLOCK
011730  001006                                  BNE·    10$                     ;IF·NOT, A TRUE·ERROR·
011732·  152761  000040  000007                 BISB·   #BUFEOF,ATTR(R1)        ;SET·END·OF·FILE·FLAG·
011740  105061  000021                          CLRB·   NXTR+1(R1)              ;RESET·I/O·IN·PROGRESS·FLAG·
011744  000412·                                 BR      BUFIOX·                 ;EXIT·ROUTINE·
011746                                  10$:
011746  016067  000052  176220                 MOV·    F.ERR(R0),IOSTAT·
011754  012700  010200'                         MOV·    #ERR1,R0
011760                                          CALL·   COOUT·
011764                                          EXIT$S·
                                        ;
                                        ;       RETURN·POINT FOR·BUFIO SUBROUTINE·
                                        ;
011772·                                 BUFIOX:
011772·                                         RESTOR· R0,R2,R3,R4
012002·  000207                                 RETURN·
         000001                                 .END·
```

SYMBOL TABLE


```
ASTX    011132R          B.FEMB  000142          010 ERROR   010676R          FF.NV = 000003          F.MBCT= 000054
ATTR    000007      014  B.FEMC  000152          010 ERRX    011722R          FF.POE= 000002          F.MBC1= 000055
BDB     000164R          B.FFSA  000202          010 ERR1    010200R          FF.RWD= 000001          F.MBFG= 000056
BDB1    000164R          B.FFSB  000212          010 ERR1L = 000021          FF.RWF= 000006          F.NRBD= 000024
BDB2    000166R          B.FFSC  000222          010 ERR1T   010202R          FF.SPC= 000004          F.NREC= 000030
BITVAL= 000000           B.FMHR  000172          010 ERR2    010224R          FN.DBR  000026      011 F.OVBS= 000030
BIT0  = 000001           B.FQLS  000162          010 ERR2L = 000030          FN.DBS  000022      011 F.RACC= 000016
BIT1  = 000002           B.FSAZ  000100          010 ERR2T   010226R          FN.DHR  000040      011 F.RATT= 000001
BIT10 = 002000           B.FSBZ  000102          010 ERR3    010256R          FN.EMA  000012      011 F.RCNM= 000034
BIT11 = 004000           B.FSCZ  000104          010 ERR3L = 000017          FN.EMB  000014      011 F.RCTL= 000017
BIT12 = 010000           B.HBLK  000120          010 ERR3T   010260R          FN.EMC  000016      011 F.RSIZ= 000002
BIT13 = 020000           B.HDOC  000114          010 EVNT    000006      014 FN.FSA  000000      011 F.RTYP= 000000
BIT14 = 040000           B.HRLP  000126          010 EXIT    010702R          FN.FSB  000002      011 F.SEQN= 000100
BIT15 = 100000           B.HRLR  000122          010 FA.APD= 000100           FN.FSC  000004      011 F.SPDV= 000072
BIT2  = 000004           B.HRLW  000124          010 FA.CRE= 000010           FN.LGO  000034      011 F.SPUN= 000074
BIT3  = 000010           B.NMBR  000052          010 FA.DLK= 001000           FN.LGU  000036      011 F.STBK= 000036
BIT4  = 000020           B.NQRY  000232          010 FA.ENB= 100000           FN.MFO  000024      011 F.UNIT= 000136
BIT5  = 000040           B.QLSZ  000106          010 FA.EXC= 002000           FN.MHR  000010      011 F.URBD= 000020
BIT6  = 000100           B.QMAP  000234          010 FA.EXT= 000004           FN.NMB  000044      011 F.VBN = 000064
BIT7  = 000200           B.QSPL  000316          010 FA.NSP= 000100           FN.QLS  000006      011 F.VBSZ= 000060
BIT8  = 000400           B.QTTM  000076          010 FA.POS= 010000           FN.QRY  000020      011 GETBUF  011002R
BIT9  = 001000           B.QUQP  000056          010 FA.RD = 000001           FN.SF0  000030      011 GTBLK   011300R
BLDEFL= ****** G         B.SFDB  000010          010 FA.RWD= 004000           FN.SF1  000032      011 IBDB    000022      014
BSTADR  010170R          B.SIZE  000772          010 FA.SEQ= 040000           FN.SHD  000042      011 IE.ABO= ****** GX
BSTPTR= ****** G         B.SNDP  000012          010 FA.SHR= 000040           FO.APD= 000106          IE.CNR= ****** GX
BS.CLS= 000002           B.SSQ   000004          010 FA.TMP= 000020           FO.MFY= 000002          IE.DNR= ****** GX
BS.DBU= 000004           B.SSQF  000050          010 FA.WCK= 020000           FO.RD = 000001          IE.EOF= ****** GX
BS.INA= 000000           B.STAT  000044          010 FA.WRT= 000002           FO.UPD= 000006          IE.TMO= ****** GX
BS.OPN= 000001           B.STTE  000053          010 FDB     000000R          FO.WRT= 000016          IOCB    000140R
BS.SRC= 000003           B.UDOC  000110          010 FDBAD   000004      014 F.ACTL= 000076          IOST    000000      014
BUFEOF= 000040           CBLK    000010          014 FD.BLK= 000010           F.ALOC= 000040          IOSTAT  010174R
BUFIO   011316R          CBUF    000014          014 FD.CCL= 000002           F.BBFS= 000062          IO.WLB= ****** GX
BUFIOX  011772R          CF.B0 = 000070               FD.COM= 020000           F.BDB = 000070          IS.SUC= ****** GX
BUFLOK= 000001           CF.B2 = 000067               FD.CR = 000002           F.BGBC= 000057          LBDB    000023      014
BUFOPN= 000200           CF.B4 = 000066               FD.DIR= 000010           F.BKDN= 000026          LBLK    000012      014
BUFQIO  011452R          CF.B6 = 000065               FD.FID  000000      003 F.BKDS= 000020          LDHRL   010300RG
BUFRAP= 000020           CF.DR0= 000064               FD.FNB  000006      003 F.BKEF= 000050          LDXIT   010762R
BUFRD = 000004           CF.DR1= 000063               FD.FTN= 000001           F.BKP1= 000051          LNKRST  010640R
BUFWR = 000010           CH.AND= 000001               FD.FVR  000004      003 F.BKST= 000024          M       000062
BUFXIT  011126R          CLOFIL  010626R              FD.Fil= 040000           F.BKVB= 000064          M.KTAE= 000010
BUF1    000170R          COLUN = 000006               FD.INS= 000010           F.CHR = 000075          M.KTEF= 000002
BUF2    004170R          COOUT   010710R              FD.ISP= 002000           F.CNTG= 000034          M.KTMG= 000004
BYTE0 = 000000           DBSLEN= 000116               FD.LEN  000010      003 F.DFNB= 000046          M.KTUN= 000006
BYTE1 = 000001           DH.BF0  000002          005 FD.MNT= 100000           F.DSPT= 000044          N     = 000002
BYTE2 = 000002           DH.BF1  000004          005 FD.OSP= 004000           F.DVNM= 000134          NB.DEV= 000200
BYTE3 = 000003           DH.CTL  000000          005 FD.PLC= 000004           F.EFBK= 000000          NB.DVR= 000100
BYTE4 = 000004           DH.DMC  000010          005 FD.PRN= 000004           F.EFN = 000050          NB.NAM= 000004
BYTE5 = 000005           DH.FLG  000006          005 FD.PSE= 010000           F.EOBB= 000032          NB.SD1= 000400
BYTE6 = 000006           DN.DCK  000000          013 FD.RAH= 000001           F.ERR = 000052          NB.SD2= 001000
BYTE7 = 000007           DN.NTP  000004          013 FD.RAN= 000002           F.FACC= 000043          NB.SNM= 000040
BYTE8 = 000010           DN.NXT  000006          013 FD.REC= 000001           F.FFBY= 000014          NB.STP= 000020
BYTE9 = 000011           DN.ROT  000002          013 FD.RWM= 000001           F.FNAM= 000110          NB.SVR= 000010
BYTVAL= 000012           DN.SIZ  000010          013 FD.SDI= 000020           F.FNB = 000102          NB.TYP= 000002
B.BSTA  000054      010  DPLUN = 000003               FD.SQD= 000040           F.FTYP= 000116          NB.VER= 000001
B.CNTX  000046      010  DSKERR  010464R              FD.TTY= 000004           F.FVER= 000120          NXTCTL  000016      014
B.COUO  000060      010  DSKOK   010472R              FD.WBH= 000002           F.HIBK= 000004          NXTR    000020      014
B.FEMA  000132      010  EF.IO = 000001               FF.CHR= 000005           F.LUN = 000042          N.BFAC= 000004
```

```
LDHRL  M[illegible]O M1110  27-MAR-80 13:20  PAGE 19-2
SYMBOL TA[illegible]
```

```
N.BHGH= 000006          R.VAR = 000002          SR.1IP  000016      002 S.FNAM= 000006          XBATCH= 000013
N.BTCH= 000004          RiSAV   010172R         SS.FID  000002      004 S.FNB = 000036          XDBLOA= 000004
N.BUFB= 004000          SR.ARE  000114      002 SS.FNB  000010      004 S.FNBW= 000017          XDBPRO= 000012
N.BUFW= 002000          SR.ARS  000106      002 SS.FVR  000006      004 S.FNTY= 000004          XDMCIN= 000006
N.DID = 000024          SR.DAY  000010      002 SS.LEN  000012      004 S.FTYP= 000002          XFOSMR= 000007
N.DVNM= 000032          SR.DLT  000014      002 SS.STT  000000      004 S.HRL = 000240          XGTSRE= 000014
N.FID = 000000          SR.ECB  000047      002 ST.ASZ  000020      006 S.NFEN= 000020          XHITSK= 000011
N.FNAM= 000006          SR.ECH  000046      002 ST.BSZ  000024      006 TOPLUP  010426R         XHLMER= 000002
N.FOS = 000764          SR.ECL  000050      002 ST.BTC  000000      006 TSTRED  010454R         XHOTSK= 000010
N.FTYP= 000014          SR.FIB  000012      002 ST.CSZ  000030      006 WN.NTP  000004      012 XMLUN = ****** GX
N.FVER= 000016          SR.GRE  000100      002 ST.HRL  000010      006 WN.NXT  000006      012 XMSCHE= 000000
N.NEXT= 000022          SR.GRS  000072      002 ST.LEN  000044      006 WN.ROT  000002      012 XQTS  = 000003
N.PKSZ= 000020          SR.LEN  000122      002 ST.QRY  000002      006 WN.SIZ  000010      012 XQT0  = 000001
N.PKTS= 000043          SR.LIN  000066      002 ST.QSZ  000034      006 WN.SRC  000000      012 XSULOA= 000005
N.QURY= 000031          SR.LIP  000062      002 ST.SCH  000040      006 WN.TYP  000001      012 $$$   = 000012R     016
N.STAT= 000020          SR.MON  000006      002 ST.UHL  000004      006 WORD0 = 000000          $$$ARG= 000002
N.SUNT= 000002          SR.NDC  000042      002 ST.XLT  000014      006 WORD1 = 000002          $$$OST= 000004
N.UNIT= 000034          SR.NDS  000036      002 SU.DBU= 000004          WORD2 = 000004          .CLOSE= ****** G
PAR$$$= 000061          SR.NIN  000030      002 SU.DON= 000006          WORD3 = 000006          .FSRCB= ****** G
QE.ROI= 000144          SR.NIP  000022      002 SU.IDL= 000000          WORD4 = 000010          .OPFNB= ****** G
Q.FDSC  000004      007 SR.SDB  000032      002 SU.LOD= 000001          WORD5 = 000012          .READ = ****** G
Q.NOBK  000000      007 SR.SRC  000002      002 SU.SRC= 000002          WORD6 = 000014          .WAIT = ****** G
Q.NUHL  000002      007 SR.SUN  000000      002 SU.SRR= 000005          WORD7 = 000016          .WRITE= ****** G
Q.SIZE  000014      007 SR.TWS  000056      002 SU.XPD= 000003          WORD8 = 000020          ...GBL= 000000
REDFIL  010444R         SR.USL  000052      002 S.BFHD= 000020          WORD9 = 000022          ...PC1= 000000R
RELBUF  011060R         SR.YR   000004      002 S.FATT= 000016          WRDVAL= 000024          ...PC2= 000140R
R.FIX = 000001          SR.1IN  000024      002 S.FDB = 000140          W.TSEF= 000002          ...TPC= 000020
R.SEQ = 000003
```

```
. ABS.  000000     000
        012004     001
SRCOFF  000122     002
FDSCOF  000010     003
SUSOFF  000012     004
DHROFF  000012     005
STTOFF  000044     006
QSPLOF  000014     007
BSTOFF  000772     010
FNOFFS  000044     011
WNODOF  000010     012
DNODOF  000010     013
IOCBOF  000024     014
$$FSR1  000000     015
$DPB$$  000016     016
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  7245 WORDS  (29 PAGES)
DYNAMIC MEMORY:  8084 WORDS  ( 31 PAGES)
ELAPSED TIME:  00:00:55
LDHRL,LDHRL/-SP=[20,1]P,M,LDHRL
```

11- 21 HRLMRG MAIN ROUTINE
12- 71 HLMERG - MERGE UHL COMMAND
16- 218 HLTERM - TERMINATE HRL PROCESSING FOR THIS BATCH
17- 226 HLABRT - ABORT HRL PROCESSING FOR BATCH
18- 246 HLPROB - PROBE UHL SIZE SUBROUTINE
20- 322 LOCATE BATCH TABLE AND QUERY SPOOL FILE SUBROUTINE
21- 357 INITIALIZE HRL MERGE FILE SUBROUTINE
23- 456 START UP UHL FILE SUBROLTINE (ULINIT)
24- 495 GET NEXT HRL ENTRY ROUTINE (HGET)
25- 535 GET NEXT UHL ENTRY ROUTINE (UGET)
26- 577 OUTPUT AN HRL ENTRY SUBROUTINE (HPUT)
27- 623 BUFFER CON'ROL ROUTINES
32- 824 ERROR HANDLING ROUTINE
33- 876 DATA STORAGE AND SYMBOLIC VARIABLES
34- 934 FILE STRUCTURES

```
                                .TITLE  HRLMRG
                        ;
                        ;   GENERATE HIT RESOLUTION LIST (HRL) FILE PROGRAM
                        ;
                        ;   THIS PROGRAM MERGES A USER HIT LIST (UHL) INTO AN HRL FILE,
                        ;   ESSENTIALLY AN ORDERING OF DOCUMENT IDS WITH THEIR ASSOCIATED
                        ;   QUERY IDS.  THE PROGRAM LOGIC IS JUST A VARIATION OF THE CLASSIC
                        ;   OLD MASTER (HRL) & TRANSACTION (UHL) -> NEW MASTER (HRL).
                        ;
                        ;   THE PROGRAM USES A GENERALIZED MULTIPLE BUFFERING SCHEME FOR THE
                        ;   INPUT FILES TO SIMULATE THE (IDEALIZED) GET FUNCTION, WHICH DOES
                        ;   NOT HAVE A MULTIPLE BUFFERING CAPABILITY IN RSX-11M.
                        ;
000012                          .RADIX  10
                                .NLIST  BEX
                        ;
                                .MCALL  EXIT$S,RCVX$C,ASTX$S
                                .MCALL  FINIT$,READ$,WRITE$,WAIT$,OFNB$R,OFNB$W,OFNB$M,CLOSE$
                                .MCALL  FDBDF$,FDRC$A,FDBK$A,FDOP$A,FDBK$R,FSRSZ$,FDAT$R
```

```
                                        .SBTTL  HRLMRG·MAIN·ROUTINE·
                                        .PSECT·
                                HRLMRG:
                                        FINIT$
                                ;
                                ;       GET·COMMAND·PACKET·FROM·MSCHED·
                                ;
                                GETCMD:
                                        RCVX$C  ,RCVB·                  ;GET·COMMAND·DATA·
                                        BCC     10$                     ;BRANCH·IF·RECEIVED·OK·
                                        CALL    DIRERR·                 ;MSGOUT·-·DIRECTIVE·ERROR·
                                        BR      PGMXIT·                 ;EXIT·PROGRAM·
                                ;
                                ;       INTERPRET·COMMAND·-·LOCATE·BATCH·AND·QUERY
                                10$:
                                        CALL    LOCQRY·                 ;GET·CORRECT·BST·AND·QID·
                                        BCS     PGMXIT·                 ;BRANCH·IF·FAILURE·
                                        MOVB    CMDX,R2·                ;GET·COMMAND·INDEX·
                                        BMI     25$                     ;BRANCH·INVALID·INDEX·
                                        CMP     R2,#<CMDLST-CMDTAB>/2·  ;INDEX·IN·RANGE?·
                                        BGE     25$                     ;BRANCH·IF·OUT·OF·RANGE·
                                        ASL     R2·                     ;SUBROUTINE·TABLE·OFFSET·
                                ;
                                ;       EXECUTE·COMMAND·
                                ;
                                        CALL    @CMDTAB(R2)
                                        BCC     CMDLST·                 ;BRANCH·TO·ACKNOWLEDGE·SCHEDULER·
                                        BR      PGMXIT·                 ;BRANCH·IF·COMMAND·FAILURE·
                                25$:
                                        MOV     R2,PAR1                 ;STORE·COMMAND·CODE·RECEIVED·FOR·MSGOUT·
                                        MOUT$S  #MSG3,#MSGPAR·          ;INVALID·COMMAND·RECEIVED·
                                        BR      PGMXIT·
                                ;
                                ;       COMMAND·PROCESSING·ROUTINE·LIST·
                                CMDTAB:
                                        .WORD   HLMERG·                 ;MERGE·UHL·ROUTINE·
                                        .WORD   HLTERM·                 ;TERMINATE·HRL·
                                        .WORD   HLPROB·                 ;PROBE·UHL·SIZE·
                                        .WORD   HLABRT·                 ;ABORT·HRL·PROCESSING·
                                CMDLST:
                                        MOVB    #XHLMERG,PGMID          ;PROGRAM·SOURCE·=·HRLMRG·
                                        CALL    GETFRE·                 ;GET·SSQ·PACKET·-·RETURNED·IN·R2·
                                        MOV     PGMID,2(R2)             ;MOVE·IN·PROG·ID·AND·ACK·CODE·
                                        MOV     BCHNO,4(R2)             ;MOVE·IN·BATCH·ID·
                                        MOV     QRYID,6(R2)             ;MOVE·IN·QUERY·ID·
                                        CALL    PUTSSQ·                 ;CHAIN·SSQ·TO·MSCHED
                                        BR      GETCMD·                 ;CHECK·FOR·NEW·COMMAND·PACKET·
                                PGMXIT:
                                        EXIT$S·
```

```
                                        .SBTTL  HLMERG - MERGE UHL COMMAND
                                ;
000164                          HLMERG:
000164  005767  004236                  TST     NBUHL                   ;DOES CURRENT QID HAVE UHL SUBFILE
000170  001522                          BEQ     HLMERX                  ;IF NOT, NOTHING TO DO (C IS CLEARED)
000172                                  CALL    HLINIT                  ;INITIALIZE HRL FILE
000176  103517                          BCS     HLMERX                  ;OPEN FAILURE ON HRL FILE
000200                          20$:
000200                                  CALL    ULINIT                  ;START UP INPUT ON UHL SUBFILE
000204  103514                          BCS     HLMERX                  ;OPEN FAILURE ON UHL
                                ;
                                ;       OPEN HRL MERGE FILE - PRIME ITS INPUT BUFFERS
                                ;
000206                          30$:
000206                                  CALL    HGET                    ;GET NEXT HRL ENTRY FROM OLD MERGE FILE
000212                          40$:
000212                                  CALL    UGET                    ;GET NEXT UHL ENTRY
                                ;
                                ;       COMPARE UHL AGAINST INPUT HRL
000216                          50$:
000216  012703  004466'                 MOV     #UDOC,R3                ;SET POINTER TO UHL ENTRY
000222  016704  004250                  MOV     HDOC,R4                 ;SET POINTER TO HRL ENTRY
000226  012705  000003                  MOV     #3,R5                   ;LOAD DOCUMENT ID WORD LENGTH
000232                          55$:
000232  022324                          CMP     (R3)+,(R4)+             ;COMPARE A DOCUMENT ID WORD
000234  002416                          BLT     70$                     ;UHL ENTRY IS LOW
000236  003020                          BGT     80$                     ;HRL ENTRY IS LOW
000240  077504                          SOB     R5,55$                  ;CHECK NEXT WORD
                                ;
                                ;       MATCHED UHL, HRL DOCUMENT IDS
                                ;
000242  005767  004166                  TST     EOHRL                   ;END OF OLD MERGE FILE
000246  001406                          BEQ     60$                     ;BRANCH IF NOT
000250  016702  004240                  MOV     EODFLG,R2               ;LOAD END OF DATA SENTINEL
000254  005003                          CLR     R3                      ;IMMEDIATE ENTRY
000256                                  CALL    HPUT                    ;OUTPUT SENTINEL
000262  000413                          BR      90$                     ;CLEANUP MERGE COMMAND
000264                          60$:
000264                                  CALL    MRGDEQ                  ;PROCESS MATCHED ENTRIES
000270  000746                          BR      30$                     ;GET NEXT HRL AND UHL ENTRIES
                                ;
                                ;       UHL ENTRY IS LOW
000272                          70$:
000272                                  CALL    MRGDLT                  ;TRANSFER OLD UHL ENTRY
000276  000745                          BR      40$                     ;GET NEXT UHL ENTRY
                                ;
                                ;       HRL ENTRY IS LOW
000300                          80$:
000300                                  CALL    MRGDGT                  ;TRANSFER OLD HRL ENTRY
000304                                  CALL    HGET                    ;GET NEXT HRL ENTRY
000310  000742                          BR      50$                     ;GO TO COMPARE CYCLE
```

```
                                        ;       END·OF·MERGE CLEANUP·
000312·                                 90$:
000312·  012701  004520'                        MOV·    #IOCB1,R1               ;LOAD·THE·UHL·SUBFILE·IOCB·
000316                                          CALL·   RELBUF·                 ;RELEASE·LAST·BUFFER
000322·  142761  000240  000007                 BICB·   #BUFOPN!BUFEOF,ATTR(R1)  ;RESET·THE IOCB·
000330                                          CLOSE$  FDBAD(R1)               ;CLOSE·THE SPOOL·FILE·
                                        ;
000340  012701  004544'                         MOV·    #IOCB2,R1               ;LOAD·THE·HRL·INPUT·FILE·IOCB·
000344                                          CALL·   RELBUF·                 ;RELEASE·LAST·BUFFER
000350  142761  000240  000007                 BICB·   #BUFOPN!BUFEOF,ATTR(R1)  ;RESET·THE IOCB·
                                        ;
000356  012701  004570'                         MOV·    #IOCB3,R1               ;LOAD·THE·HRL·OUTPUT FILE·IOCB
000362·                                         CALL·   RELBUF·                 ;FORCE·LAST·BUFFER·TO·BE·OUTPUT·
000366                                          WAIT$   FDBAD(R1),EVNT(R1)      ;MAKE·SURE ALL·WRITE-BEHIND·COMPLETE·
000404  142761  000240  000007                 BICB·   #BUFOPN!BUFEOF,ATTR(R1)  ;RESET·THE IOCB·
000412·                                         CLOSE$                          ;CLOSE·THE HRL,MRG·FILE·
                                        ;
000416  016702·  004002                         MOV·    BSTADR,R2·              ;GET·BATCH STATUS·TABLE·
000422·  016262·  000124  000122·               MOV·    B.HRLW(R2),B.HRLR(R2)   ;SET·UP·NEW· INITIAL·READ·BLOCK
000430  016162·  000010  000124                 MOV·    CBLK(R1),B.HRLW(R2)     ;SET·UP·NEW· INITIAL·WRITE·BLOCK·
                                        ;
                                        ;       MERGE·COMMAND·EXIT·
000436                                  HLMERX:
000436  000207                                  RETURN·
```

```
                                        ;
                                        ;       MATCHED DOCUMENT ID PROCESSING SUBROUTINE.
                                        ;
000440                                  MRGDEQ:
000440  032714  000001                        BIT     #BIT0,@R4               ;ODD NUMBER OF EXISTING QIDS?
000444                                        BEQ     10$                     ;IF SO, PAD BYTE NOT AVAILABLE
000446  112403                                MOVB    (R4)+,R3                ;PICK UP QUERY COUNT
000450  042703  177400                        BIC     #^O177400,R3            ;CLEAR HIGH ORDER BITS
000454  060304                                ADD     R3,R4                   ;POINT TO PAD BYTE
000456  116714  003714                        MOVB    QRYID,@R4               ;STORE QUERY ID INTO PREVIOUS PAD BYTE
000462  016702  004010                        MOV     HDOC,R2                 ;GET START OF ENTRY
000466  105262  000006                        INCB    6(R2)                   ;BUMP COUNT OF QIDS
000472                                        CALL    HRLNTH                  ;CALCULATE HRL ENTRY WORD LENGTH
000476                                        CALL    HPUT                    ;TRANSFER MERGED ENTRY
000502  000421                              BR      30$                     ;EXIT TO GET NEXT UHL AND HRL ENTRIES
                                        ;
                                        ;       PREVIOUS QUERY COUNT WAS ODD - THIS MEANS THAT THE MERGED ENTRY
                                        ;       IS ONE WORD LONGER THAN THE OLD ENTRY.  WE TRANSFER THE ENTRY
                                        ;       IN TWO HPUT CALLS.  THE QID COUNT IS BUMPED (IN THE HRL INPUT BUFFER)
                                        ;       AND THE FIRST HPUT CALL TRANSFERS ALL EXCEPT THE NEW LAST WORD DIRECTLY
                                        ;       FROM THE INPUT BUFFER.  THE SECOND HPUT CALL IS AN IMMEDIATE TRANSFER
                                        ;       OF THE CURRENT QID (I.E., QRYID LOADED INTO THE LOW ORDER BYTE OF R3).
                                        ;       THERE IS, HOWEVER, ONE PITFALL TO BE AVOIDED.  IF THE OLD ENTRY WOULD
                                        ;       HAVE EXACTLY FIT THE REMAINING SPACE IN THE MRG OUTPUT BUFFER, TO
                                        ;       TRANFER THE ENTRY IN TWO HPUT CALLS WOULD SPLIT IT ACROSS BLOCK BOUNDARIES.
                                        ;       TO MAKE SURE THAT THIS DOES NOT HAPPEN, SCOUNT(THE AMOUNT OF SPACE REMAINING
                                        ;       IN THE OUTPUT BUFFER) IS DECREMENTED BEFORE THE FIRST HPUT CALL.
                                        ;
000504                                  10$:
000504  016702  003766                        MOV     HDOC,R2                 ;GET START OF DOCUMENT
000510                                        CALL    HRLNTH                  ;CALCULATE WORD LENGTH OF OLD ENTRY
000514  105262  000006                        INCB    6(R2)                   ;BUMP QID COUNT
000520  005367  003732                        DEC     SCOUNT                  ;ALLOCATE POTENTIAL SPACE FOR 2ND HPUT
000524                                        CALL    HPUT                    ;MOVE FIRST SEGMENT OF MERGED ENTRY
000530  005267  003722                        INC     SCOUNT                  ;CORRECT THE SPACE COUNT
000534  016702  003636                        MOV     QRYID,R2                ;SET UP SECOND SEGMENT OF MERGED ENTRY
000540  005003                                CLR     R3                      ;IMMEDIATE ENTRY TRANSFER
000542                                        CALL    HPUT                    ;MOVE SECOND SEGMENT
000546                                  30$:
000546  000207                              RETURN
```

```
                                        ;       TRANSFER·OLD UHL·SUBROUTINE·
                                        ;
000550                                  MRGDLT:
000550  012702· 004466'                         MOV·    #UDOC,R2·
000554  116767  003616  003713                  MOVB·   QRYID,UDOC+7
000562· 012703  000004                         MOV·    #4,R3
000566                                         CALL·   HPUT
000572· 000207                                 RETURN·
                                        ;
                                        ;
                                        ;       TRANSFER·OLD HRL·SUBROUTINE·
                                        ;
000574                                  MRGDGT:
000574  016702· 003676                         MOV·    HDOC,R2·
000600                                         CALL·   HRLNTH·
000604                                         CALL·   HPUT
000610  000207                                 RETURN·
                                        ;
                                        ;
                                        ;       CALCULATE·WORD·SIZE·OF (MERGED) HRL·ENTRY·SUBROUTINE·
                                        ;       (R2) =·ADDRESS·OF·START·WORD·OF·ENTRY·
                                        ;       ((R2)+6) =·NUMBER·OF·QUERY·IDS·IN·THE·ENTRY·
                                        ;
000612·                                 HRLNTH:
000612·  116203  000006                         MOVB·   6(R2),R3                ;GET·QID·COUNT·
000616  062703  000010                         ADD·    #8.,R3                  ;ALLOW·FOR DOC·ID, COUNT, AND·PAD·
000622· 006203                                 ASR·    R3                      ;CONVERT·TO·WORD·
000624  000207                                 RETURN·
```

```
218·
219                                             .SBTTL· HLTERM·- TERMINATE·HRL·PROCESSING·FOR·THIS·BATCH·
220 000626                                      ;
221 000626  016703  003572                      HLTERM:
222·000632· 005063  000124                              MOV·    BSTADR,R3               ;GET·CURRENT·BATCH·STATUS·TABLE·
223 000636  000241                                      CLR·    B.HRLW(R3)              ;RESET·HRL FILE·INITIALIZED·FLAG·
224 000640  000207                                      CLC·                            ;SET·NORMAL·RETURN·CODE·
                                                        RETURN·
```

```
                                        .SBTTL  HLABRT -- ABORT HRL PROCESSING FOR BATCH
                                ;
000642                          HLABRT:
000642  016703  003556                  MOV     BSTADR,R3               ;GET CURRENT BATCH STATUS TABLE
000646  005763  000124                  TST     B:HRLW(R3)              ;HAS HRL:MRG BEEN INITIALIZED
000652  001420                          BEQ     20$                     ;EXIT IF SO
000654  005267  003604                  INC     ABRTFL                  ;SET ABORT FLAG
000660                                  CALL    HLINIT                  ;INITIALIZE HRL FILE
000664  016002  000004                  MOV     F:HIBK(R0),R2           ;GET ORIGINAL HRL FILE ALLOCATION
000670  006202                          ASR     R2                      ;CONVERT TO LOGICAL BLOCKS
000672  006202                          ASR     R2
000674  010263  000120                  MOV     R2,B.HBLK(R3)           ;RESTORE ORIGINAL HRL FILE SIZE
000700  005063  000124                  CLR     B:HRLW(R3)              ;RESET HRL FILE INITIALIZED FLAG
000704  005067  003554                  CLR     ABRTFL                  ;CLEAR ABORT FLAG
000710                                  CALL    .DLFNB                  ;DELETE THE HRL FILE
000714                          20$:
000714  105267  003453                  INCB    CMDX                    ;SET UP ACKNOWLEDGE CODE
000720  000241                          CLC                             ;SET NORMAL RETURN CODE
000722  000207                          RETURN
```

```
                                        .SBTTL  HLPROB - PROBE UHL SIZE SUBROUTINE
                                ;
000724                          HLPROB:
000724  005767  003476                TST     NBUHL                   ;DOES CURRENT QID HAVE UHL SUBFILE?
000730  001530                        BEQ     HLPRBX                  ;IF NOT, AUTOMATIC SUCCESS OF PROBE
000732  016703  003466                MOV     BSTADR,R3               ;GET CURRENT BATCH STATUS TABLE
000736  005763  000124                TST     B.HRLW(R3)              ;HAS HRL.MRG ALREADY BEEN INITIALIZED?
000742  001022                        BNE     10$                     ;IF SO, WE NEED NOT OPEN HRL FILE
000744                                CALL    HLINIT                  ;INITIALIZE HRL FILE
000750  012701  004570'               MOV     #IOCB3,R1               ;LOAD THE IOCB AGAIN
000754                                CALL    RELBUF                  ;FORCE LAST BUFFER TO BE OUTPUT
000760  005363  000120                DEC     B.HBLK(R3)              ;REDUCE AVAILABLE HRL BLOCK COUNT
000764                                WAIT$   ,EVNT(R1)               ;MAKE SURE ALL WRITE-BEHIND COMPLETE
000776  142761  000240  000007        BICB    #BUFOPN!BUFEOF,ATTR(R1) ;RESET THE IOCB
001004                                CLOSE$                          ;CLOSE THE HRL.MRG FILE
                                ;
                                ;       APPLY BLOCK TEST. IF ((NBUHL*4)+2)/3 <= B.HBLK, THE UHL SUBFILE
                                ;       WILL DEFINITELY FIT. THUS, IF THIS CONDITION IS TRUE, WE DO NOT
                                ;       READ THE UHL SUBFILE TO DETERMINE THE ACTUAL NUMBER OF ENTRIES.
                                ;
001010                          10$:
001010  016701  003412                MOV     NBUHL,R1                ;GET NUMBER OF UHL BLOCKS
001014  006301                        ASL     R1
001016  006301                        ASL     R1                      ;MULTIPLY BY 4
001020  062701  000002                ADD     #2,R1
001024  005000                        CLR     R0                      ;SET UP AND PERFORM DIVIDE
001026  071027  000003                DIV     #3,R0                   ;END OF REQUIRED HRL BLOCKS CALCULATION
001032  016703  003366                MOV     BSTADR,R3               ;GET CURRENT BATCH STATUS TABLE
001036  020063  000120                CMP     R0,B.HBLK(R3)           ;COMPARE WITH REMAINING HRL BLOCK COUNT
001042  003462                        BLE     20$                     ;UHL WILL DEFINITELY FIT
                                ;
                                ;       AT THIS POINT, IF WE REDUCE BY 1 THE REQUIRED NUMBER OF
                                ;       HRL BLOCKS, AND THERE ARE STILL NOT ENOUGH HRL BLOCKS LEFT,
                                ;       OBTAINING THE ACTUAL ENTRY COUNT IS FUTILE, SINCE THE ESTIMATED
                                ;       REQUIRED BLOCKS (AS CALCULATED FROM AN ENTRY COUNT) WOULD FAIL
                                ;       ANYWAY; THUS HERE TOO, THERE IS NO NEED TO OPEN THE UHL SUBFILE.
                                ;
001044  005300                        DEC     R0
001046  020063  000120                CMP     R0,B.HBLK(R3)
001052  003054                        BGT     18$                     ;REPORT NO FIT ESTIMATE
```

```
                                        ;       THE·CASE·WE·HAVE·NOW·IS·THAT·THE·UHL·FILE·MAY·POSSIBLY·FIT,
                                        ;       DEPENDING·UPON·THE·ACTUAL·NUMBER·OF·ENTRIES·IN·THE·LAST·BLOCK.
                                        ;       SO·WE·WILL·PROCEED·TO·READ·THE·SUBFILE·HEADER·TO·EXTRACT·THE·
                                        ;       ACTUAL·UHL·ENTRY·COUNT.
                                        ;
001054  012702·  004520'               MOV·    #IOCB1,R2·              ;IOCB·OF·UHL·SUBFILE
001060  016762·  003344  000012·        MOV·    STUHL,LBLK(R2)          ;PRIME·ONLY·A·SINGLE·BUFFER·
001066                                 CALL·   ULINIT·                 ;GET·FIRST·BLOCK·OF·UHL·
001072·  103015                        BCC·    12$                     ;BRANCH·IF·OK·
001074  016767  003276  003276         MOV·    QRYID,PAR1              ;MSGOUT·PARAMETER·
001102·                                MOUT$S· #MSG2,#MSGPAR·          ;OPEN·FAILURE·-·UHL·SUBFILE·
001122·  000261                        SEC·                            ;SET·RETURN·CODE·TO·FAILURE·
001124  000432·                        BR      HLPRBX·                 ;EXIT·SUBROUTINE·
001126                          12$:
001126  012701  004520'               MOV·    #IOCB1,R1               ;LOAD·THE·UHL·SUBFILE·IOCB·
001132·                                CALL·   RELBUF·                 ;RELEASE·LAST·BUFFER
001136  142761  000240  000007         BICB·   #BUFOPN!BUFEOF,ATTR(R1) ;RESET·THE·IOCB·
001144                                 CLOSE$  FDBAD(R1)               ;CLOSE·THE·SPOOL·FILE·
                                        ;
001154  005767  003256                 TST·    UHLCNT·                 ;CHECK·UHL·ENTRY·COUNT·
001160  001413                        BEQ·    20$                     ;IF·NONE, AUTOMATIC·FIT·
001162·  116701  003251                MOVB·   UHLCNT+1,R1             ;EACH·GROUP·OF·256·UHL·ENTRIES
001166  105767  003244                 TSTB·   UHLCNT·                 ;POTENTIALLY·REQUIRES·A·BLOCK·
001172·  001401                        BEQ·    15$                     ;IN·THE·MERGED·HRL·FILE·
001174  005201                        INC·    R1                      ;IN·THE·WORST·CASE·OF·NO·DOC·ID·OVERLAP·
001176                          15$:
001176  020163  000120                CMP·    R1,B.HBLK(R3)           ;COMPARE·ESTIMATED·BLOCKS·WITH·ACTUAL·
001202·  003402·                       BLE·    20$
001204                          18$:
001204  105267  003163                INCB·   CMDX
001210                          20$:
001210  000241                         CLC·
001212·                         HLPRBX:
001212·  000207                        RETURN·
```

```
                                        .SBTTL  LOCATE BATCH TABLE AND QUERY SPOOL FILE SUBROUTINE
                                ;
                                ;       THIS ROUTINE INTERPRETS THE COMMAND PACKET SENT BY MSCHED
                                ;       IT LOCATES AND VERIFIES THE CORRECT BATCH STATUS TABLE TO BE USED
                                ;       IT LOCATES THE QUERY SPOOL FILE DATA WITHIN THE BST, EXTRACTS THE
                                ;       START BLOCK OF THE UHL SUBFILE AND THE NUMBER OF UHL BLOCKS, AND
                                ;       EXITS WITH R1 POINTING TO THE FDSC OF THE QUERY SPOOL FILE
                                ;
001214                          LOCQRY:
001214  105067  003155                  CLRB    BCHNO+1
001220  016702  003150                  MOV     BCHNO,R2                ;GET COMMAND BATCH NUMBER
001224  016203  000000G                 MOV     BSTPTR(R2),R3           ;GET ADDRESS OF BST
001230  010367  003170                  MOV     R3,BSTADR               ;SAVE IT
001234  132767  000001  003131          BITB    #1,CMDX                 ;IS THIS ABORT OR TERMINATE COMMAND?
001242                                  BON     LOCXIT                  ;EXIT IF SO
001244  016701  003126                  MOV     QRYID,R1                ;CHECK QUERY ID IN COMMAND PACKET
001250  003015                          BGT     20$                     ;BRANCH IF VALID AND ID > 0
001252  001416                          BEQ     25$                     ;BRANCH IF VALID AND ID = 0
001254  010167  003120                  MOV     R1,PAR1                 ;STORE QUERY ID AS MSGOUT PARAMETER
001260                                  MOUT$S  #MSG5,#MSGPAR           ;INVALID QUERY ID
001300  000261                          SEC                             ;SET FAILURE RETURN CODE
001302  000415                          BR      LOCXIT
001304                          20$:
001304  070127  000014                  MUL     #Q.SIZE,R1              ;OFFSET TO QUERY ENTRY
001310                          25$:
001310  062701  000316                  ADD     #B.QSPL,R1              ;DISPLACEMENT IN BST
001314  060301                          ADD     R3,R1                   ;ADDRESS OF SPOOL FILE DATA IN BST
001316  012167  003106                  MOV     (R1)+,STUHL             ;FETCH NUMBER OF QUERY BLOCKS
001322  012167  003100                  MOV     (R1)+,NBUHL             ;FETCH NUMBER OF UHL BLOCKS
001326  010167  003100                  MOV     R1,FDUHL                ;SAVE FDSC ADDRESS OF QUERY SPOOL FILE
001332  005267  003072                  INC     STUHL                   ;POINT TO START OF UHL BLOCK
001336                          LOCXIT:
001336  000241                          CLC                             ;SET SUCCESS RETURN CODE
001340  000207                          RETURN
```

```
                                        .SBTTL  INITIALIZE HRL MERGE FILE SUBROUTINE
                                ;
                                ;       THIS SUBROUTINE OPENS THE HRL FILE USING IOCB3 (OUTPUT OPERATIONS).
                                ;       SPACE FOR THE HRL FILE IS ALLOCATED (AS DETERMINED BY B.HBLK),
                                ;       IF THE HRL FILE HAS NOT BEEN INITIALIZED YET IN THIS BATCH CYCLE,
                                ;       AN END OF FILE SENTINEL IS WRITTEN IN THE FIRST BLOCK, AND
                                ;       THE COMMON FIELDS B.HRLR AND B.HRLW ARE INITIALIZED.
                                ;       THE PROGRAM ALSO SETS THE END OF HRL DATA FLAG AND POINTS TO THE
                                ;       DUMMY DOCUMENT ID SO THAT THE MERGE LOGIC CAN BE INDIFFERENT TO
                                ;       WHETHER IT IS WORKING WITH THE FIRST OR A LATER QUERY.
                                ;       IF THE MERGE FILE HAS BEEN INITIALIZED ON A PREVIOUS CALL, THE
                                ;       PROGRAM STARTS PRIMING INPUT HRL BUFFERS (USING IOCB2) WITH OLD
                                ;       MASTER HRL DATA, ASSIGNS A BUFFER TO THE PROGRAM FOR THE NEW
                                ;       MASTER FILE, AND TRANSFERS A HEADER RECORD TO THE OUTPUT BUFFER.
                                ;
001342                          HLINIT:
001342                                  SAVE    R3
001344  012700  004570'                  MOV     #IOCB3,R0               ;OUTPUT IOCB FOR HRL MERGE FILE
001350  016000  000004                  MOV     FDBAD(R0),R0            ;GET ITS FDB
001354  016702  003044                  MOV     BSTADR,R2               ;GET ADDRESS OF BST
001360  005762  000124                  TST     B.HRLW(R2)              ;HAS THE FILE BEEN ALLOCATED YET?
001364  001060                          BNE     15$                     ;BRANCH IF SO
001366  012701  000010                  MOV     #FN.MHR,R1              ;FILE NUMBER OF HRL.MRG
001372                                  CALL    BLDNFL                  ;CONSTRUCT FILE NAME BLOCK IN THE FDB
001376  016203  000120                  MOV     B.HBLK(R2),R3           ;GET REQUIRED SPACE FOR HRL FILE
001402  006303                          ASL     R3                      ;CONVERT TO VIRTUAL BLOCKS
001404  006303                          ASL     R3
001406                                  FDAT$R  R0,,,,R3                ;ALLOCATE SPACE
001412                                  OFNB$W                          ;OPEN FILE
001424  103012                          BCC     10$                     ;BRANCH IF OPEN WAS OK
001426                                  MOUT$S  #MSG1,#MSGPAR           ;REPORT ERROR IF OPEN FAILED
001446  000261                          SEC                             ;SET FAILURE CONDITION CODE
001450  000542                          BR      HLNITX                  ;EXIT SUBROUTINE
001452                          10$:
001452  005262  000124                  INC     B.HRLW(R2)              ;SET UP HRL.MRG WRITE POINTER
001456  016262  000124  000122          MOV     B.HRLW(R2),B.HRLR(R2)   ;SET UP HRL.MRG READ POINTER
001464  012701  000172                  MOV     #B.FMHR,R1              ;GET ADDRESS OF THE FDSC FOR HRL.MRG
001470  060201                          ADD     R2,R1
001472  016021  000102                  MOV     F.FNB+N.FID(R0),(R1)+   ;SET UP FILE DESCRIPTOR
001476  016021  000104                  MOV     F.FNB+N.FID+2(R0),(R1)+ ;FILE ID
001502  016021  000120                  MOV     F.FNB+N.FVER(R0),(R1)+  ;VERSION NUMBER
001506  012711  000010                  MOV     #FN.MHR,(R1)            ;FILE NUMBER
001512  005267  002716                  INC     EOHRL                   ;SET END OF MASTER FILE FLAG
001516  012767  004500' 002752          MOV     #EOFDOC,HDOC            ;SET UP DUMMY DOC ID FOR HRL FILE
001524  000463                          BR      30$                     ;BRANCH TO PUT HEADER RECORD
```

```
                                        ;       SET UP MULTIPLE BUFFERING OF INPUT HRL
                                        ;
001526                                  15$:
001526  012701  000172                          MOV     #B,FMHR,R1              ;GET ADDRESS OF THE FDSC FOR HRL.MRG
001532  060201                                  ADD     R2,R1
001534                                          CALL    BLDEFL                  ;BUILD FILE NAME BLOCK IN FDB
001540                                          OFNB$M                          ;OPEN FILE
001552  103012                                  BCC     20$                     ;BRANCH IF OPEN WAS OK
001554                                          MOUT$S  #MSG1,#MSGPAR           ;REPORT ERROR IF OPEN FAILED
001574  000261                                  SEC                             ;SET FAILURE CONDITION CODE
001576  000467                                  BR      HLNITX                  ;EXIT SUBROUTINE
001600                                  20$:
001600  005767  002660                          TST     ABRTFL                  ;ABORT COMMAND?
001604  001064                                  BNE     HLNITX                  ;EXIT IF SO
001606  012701  004544'                        MOV     #IOCB2,R1               ;GET INPUT HRL IOCB
001612  016261  000122  000010                 MOV     B,HRLR(R2),CBLK(R1)     ;DEFINE LOGICAL START OF FILE
001620  016261  000124  000012                 MOV     B,HRLW(R2),LBLK(R1)     ;DEFINE LOGICAL END OF FILE(WRAPAROUND)
001626  005067  002602                          CLR     EOHRL                   ;RESET END OF HRL FILE FLAG
001632                                          CALL    BUFIO                   ;START PRIMING INPUT BUFFERS
001636                                          CALL    GETBUF                  ;ASSIGN FIRST BUFFER TO PROGRAM
001642  016100  000014                          MOV     CBUF(R1),R0             ;GET BUFFER ADDRESS
001646  026720  002636                          CMP     HRLHDR,(R0)+            ;VERIFY PRESENCE OF FILE HEADER
001652  001402                                  BEQ     25$                     ;BRANCH IF OK
                                        ;
                                        ;       IF WE GET HERE, SOMEONE HAS SCREWED UP THE POINTERS IN COMMON
                                        ;       OR THERE IS SOME SORT OF TIMING BUG
                                        ;
001654  000261                                  SEC
001656  000437                                  BR      HLNITX
                                        ;
001660                                  25$:
001660  005720                                  TST     (R0)+                   ;SKIP PAST PAD WORD
001662  010067  002560                          MOV     R0,HRLNXT               ;SET UP FOR HGET CALLS
001666  012767  003774  002556                 MOV     #N.BUFB-4,HRLCNT        ;INITIALIZE REMAINING BUFFER SPACE
                                        ;
                                        ;       SET UP WRITE BEHIND OUTPUT FOR NEW MERGE FILE (WITH WRAP AROUND)
                                        ;
001674                                  30$:
001674  012701  004570'                        MOV     #IOCB3,R1               ;GET IOCB FOR NEW HRL FILE
001700  016261  000124  000010                 MOV     B,HRLW(R2),CBLK(R1)     ;INITIALIZE LOGICAL START OF FILE
001706                                          CALL    GETBUF                  ;ASSIGN OUTPUT BUFFER TO THE PROGRAM
001712  016167  000014  002530                 MOV     CBUF(R1),HRLOUT         ;INITIALIZE BUFFER POINTER
001720  012767  002000  002530                 MOV     #N.BUFW,SCOUNT          ;INITIALIZE AVAILABLE BUFFER SPACE
001726  012702  004510'                        MOV     #HRLHDR,R2              ;LOCATE HRL FILE HEADER AND EOD SENTINEL
001732  012703  000003                          MOV     #3,R3                   ;THREE WORDS OF DATA
001736                                          CALL    HPUT                    ;OUTPUT THE HEADER RECORD
001742  005267  002510                          INC     SCOUNT                  ;FORCE NEXT HPUT(IF ANY) TO WRITE OVER EOD SENTINEL
001746  162767  000002  002474                 SUB     #2,HRLOUT               ;ADJUST BUFFER POINTER ACCORDINGLY
001754  000241                                  CLC                             ;SET SUCCESS RETURN CODE
001756                                  HLNITX:
001756                                          RESTOR  R3
001760  000207                                  RETURN
```

```
                                        .SBTTL  START UP UHL FILE SUBROUTINE (ULINIT)
                                ;
                                ;       THIS SUBROUTINE INITIALIZES THE USER HIT FILE FOR INPUT.
                                ;       IT IS CALLED BY BOTH THE PROBE UHL AND THE MERGE UHL COMMANDS.
                                ;       IF NO USER HIT LIST IS INDICATED (BY THE NUMBER OF UHL BLOCKS
                                ;       IN THE QUERY'S BST), UHLCNT WILL BE SET TO 0, AND THE RETURN
                                ;       WILL INDICATE SUCCESS.  IF THE NUMBER OF UHL BLOCKS IS POSITIVE,
                                ;       THE QUERY SPOOL FILE WILL BE OPENED, AND THE INDICATED BUFFERS
                                ;       WILL BE PRIMED WITH INPUT.  IF THE OPEN FAILS, THE RETURN CODE
                                ;       WILL BE SET TO INDICATE FAILURE.  ON ENTRY, R1 -> THE FDSC
                                ;       OF THE QUERY SPOOL FILE, AS OBTAINED FROM THE BST.
                                ;
001762                          ULINIT:
001762  005067  002450                  CLR     UHLCNT                  ;INITIALIZE NUMBER OF UHL ENTRIES
001766  005767  002434                  TST     NBUHL                   ;CHECK NUMBER OF UHL BLOCKS
001772  003440                          BLE     20$                     ;BRANCH IF THERE ARE NONE
001774  012700  004614'                MOV     #FDB1,R0                ;LOAD FDB ADDRESS OF QUERY SPOOL FILE
002000  016701  002426                  MOV     FDUHL,R1                ;LOAD FDSC ADDRESS OF QUERY SPOOL FILE
002004                                  CALL    BLDEFL                  ;INITIALIZE FILE NAME BLOCK IN FDB
002010                                  OFNB$R                          ;OPEN THE QUERY SPOOL FILE
002022  103425                          BCS     ULNITX                  ;EXIT SUBROUTINE IF OPEN FAILURE
                                ;
002024  012701  004520'                MOV     #IOCB1,R1               ;LOAD IOCB FOR UHL FILE
002030  016761  002374  000010          MOV     STUHL,CBLK(R1)          ;POINT TO UHL SUBFILE
002036  005067  002376                  CLR     UHLNXT                  ;INITIALIZE UHL BUFFER POINNTER
002042                                  CALL    BUFIO                   ;START PRIMING THE UHL BUFFERS
002046                                  CALL    UGET                    ;GET HEADER ENTRY OF UHL FILE
                                ;
002052  026767  002430  002406          CMP     UHLHDR,UDOC             ;VALID UHL SUBFILE HEADER?
002060  001402                          BEQ     10$                     ;BRANCH IF SO
002062  000261                          SEC                             ;SET RETURN CODE TO FAILURE
002064  000404                          BR      ULNITX                  ;EXIT SUBROUTINE
002066                          10$:
002066  016767  002400  002342          MOV     UDOC+4,UHLCNT           ;PICK UP NUMBER OF UHL ENTRIES
002074                          20$:
002074  000241                          CLC                             ;SET RETURN CODE TO SUCCESS
002076                          ULNITX:
002076  000207                          RETURN
```



```
                                                .SBTTL  GET NEXT HRL ENTRY ROUTINE (HGET)
                                        HGET:
                                                SAVE
                                        10$:
002100  016704  002342                          MOV     HRLNXT,R4               ;CHECK NEXT HRL ENTRY ADDRESS
002104  001425                                  BEQ     15$                     ;IF NOT SET, GET HRL BUFFER
002106  005767  002340                          TST     HRLCNT                  ;ANY SPACE LEFT IN BUFFER?
002112  003412                                  BLE     13$                     ;BRANCH IF NOT
002114  036714  002374                          BIT     EODFLG,@R4              ;TEST FOR EOD OR EOB SENTINEL
002120  100006                                  BPL     12$                     ;BRANCH IF NOT EOD SENTINEL
002122  005267  002306                          INC     EOHRL                   ;SET END OF INPUT HRL FLAG
002126  012767  004500' 002342                  MOV     #EOFDOC,HDOC            ;POINT TO DUMMY EOF DOCUMENT ID
002134  000443                                  BR      HGETX                   ;EXIT SUBROUTINE
002136                                  12$:
002136                                          BOFF    20$                     ;BRANCH IF NOT EOB SENTINEL
002140                                  13$:
002140  012701  004544'                          MOV     #IOCB2,R1               ;HRL INPUT IOCB
002144                                          CALL    RELBUF                  ;RELEASE THE INPUT BUFFER
002150  016702  002250                          MOV     BSTADR,R2               ;GET BATCH STATUS TABLE
002154  005262  000120                          INC     B.HBLK(R2)              ;RESTORE THIS BUFFER TO POOL
002160                                  15$:
002160                                          CALL    GETBUF                  ;GET NEXT HRL BUFFER
002164  016167  000014  002254                  MOV     CBUF(R1),HRLNXT         ;SET HRLNXT TO START OF BUFFER
002172  012767  004000  002252                  MOV     #N,BUFB,HRLCNT          ;INITIALIZE REMAINING BUFFER SPACE
002200  000737                                  BR      10$                     ;RETEST HRLNXT
002202                                  20$:
002202  010403                                  MOV     R4,R3                   ;COPY POINTER TO HRL_ENTRY
002204  062703  000006                          ADD     #6,R3                   ;R3 -> QUERY COUNT FOR DOCUMENT
002210  112367  002240                          MOVB    (R3)+,QCOUNT            ;SAVE QUERY_COUNT, & POINT R3 PAST IT
002214  066703  002234                          ADD     QCOUNT,R3               ;SKIP OVER QUERY ID BYTES
002220  005203                                  INC     R3                      ;ADJUST FOR POSSIBLE PAD BYTE
002222  042703  000001                          BIC     #1,R3                   ;ALIGN ON WORD BOUNDARY
002226  010467  002244                          MOV     R4,HDOC                 ;LOCATE START OF ENTRY
002232  010367  002210                          MOV     R3,HRLNXT               ;SET LOCATION OF NEXT HRL ENTRY
002236  160403                                  SUB     R4,R3                   ;CALCULATE BYTE SIZE OF ENTRY
002240  160367  002206                          SUB     R3,HRLCNT               ;SUBTRACT IT FROM REMAINING BUFFER SPACE
002244                                  HGETX:
002244                                          RESTOR
000207                                          RETURN
```

```
                                                .SBTTL  GET NEXT UHL ENTRY ROUTINE (UGET)
                                        UGET:
                                                SAVE    R2,R3,R4
                                                MOV     #3,DCNT                 ;NUMBER OF WORDS IN DOC ID
                                                MOV     #UDOC,R3                ;START OF DOC ID BUFFER
                                                TST     UHLNXT                  ;CHECK POINTER TO NEXT SOURCE WORD
                                                BNE     20$                     ;BRANCH IF DATA LEFT IN BUFFER
                                        ;
                                        ;       GET NEW UHL BUFFER
                                        10$:
                                                MOV     #IOCB1,R1               ;LOAD IOCB OF UHL FILE
                                                CALL    GETBUF                  ;GET NEXT INPUT BUFFER
                                                MOV     CBUF(R1),UHLNXT         ;INITIALIZE SOURCE WORD POINTER TO TOP OF BUFFER
                                                MOV     #N,BUFW,UHLWD           ;INITIALIZE SIZE OF BUFFER
                                        ;
                                        ;       SET UP UHL ENTRY TRANSFER
                                        20$:
                                                MOV     UHLNXT,R2               ;SOURCE ADDRESS
                                                MOV     DCNT,R4                 ;NUMBER OF WORDS TO BE TRANSFERRED
                                                CMP     UHLWD,R4                ;REMAINING BUFFER WORD COUNT SUFFICIENT?
                                                BGE     25$                     ;BRANCH IF YES
                                                MOV     UHLWD,R4                ;LIMIT TRANSFER TO REMAINDER OF INPUT BUFFER
                                                BEQ     35$                     ;BUFFER ALREADY COMPLETELY EXHAUSTED
                                        25$:
                                                SUB     R4,DCNT                 ;COMPUTE SIZE OF NEXT PARTIAL TRANSFER
                                                SUB     R4,UHLWD                ;COMPUTE REMAINING BUFFER WORD COUNT
                                                ADD     R4,UHLNXT               ;UPDATE BUFFER POINTER
                                                ADD     R4,UHLNXT               ;BYTE OFFSET
                                        30$:
                                                MOV     (R2)+,(R3)+             ;TRANSFER A WORD OF DOCUMENT ID
                                                SOB     R4,30$                  ;NEXT WORD
                                        35$:
                                                TST     DCNT                    ;HAS ENTIRE DOCUMENT ID BEEN TRANSFERRED?
                                                BEQ     40$                     ;IF SO, EXIT
                                                MOV     #IOCB1,R1               ;IF NOT, GET IOCB OF THE UHL SUBFILE
                                                CALL    RELBUF                  ;RELEASE PREVIOUS INPUT BUFFER
                                                BR      10$                     ;GET NEW UHL BLOCK
                                        40$:
                                                DEC     UHLCNT                  ;COUNT OF REMAINING DOCS IN THIS UHL
                                                RESTOR  R2,R3,R4
                                                RETURN
```

```
                                        .SBTTL  ·OUTPUT·AN·HRL·ENTRY·SUBROUTINE·(HPUT)
                                ;
                                ;       THIS·SUBROUTINE·MOVES·AN·HRL·ENTRY·TO·AN·OUTPUT·BUFFER·
                                ;       AND·ASSIGNS·THE·BUFFERS·TO·BE·WRITTEN·AS·THEY·FILL·UP.
                                ;       (R2) =·THE·ADDRESS·OF·THE·HRL·ENTRY·TO·BE·WRITTEN.
                                ;       (R3) =·THE·SIZE·(IN·WORDS)·OF·THE·HRL·ENTRY·
                                ;       IF·(R3) =·0,·THE·CONTENTS·OF·R2·ARE·TAKEN·IMMEDIATELY·AS·
                                ;       THE·ENTRY·TO·BE·WRITTEN·
                                ;
002424                          HPUT:
002424  020367  002026                CMP·    R3,SCOUNT·              ;ENOUGH·SPACE·LEFT·IN·BUFFER?·
002430  003023                        BGT·    20$                     ;BRANCH·IF·DEFINITELY·NOT·
002432· 001412·                       BEQ·    10$                     ;MAYBE·NOT
002434  005703                        TST·    R3                      ;CHECK·SIZE·OF·HRL·ENTRY·
002436  003043                        BGT·    30$                     ;BRANCH·IF·(R2)·IS·ADDRESS·
002440  010277  002004                MOV·    R2,@HRLOUT·             ;IMMEDIATE·ENTRY·IN·R2·
002444  062767  000002  001776        ADD·    #2,HRLOUT·              ;ADJUST·BUFFER·POINTER·
002452· 005367  002000                DEC·    SCOUNT·                 ;ADJUST·SPACE·LEFT·IN·BUFFER·
002456  000443                        BR      40$
002460                          10$:
002460  005703                        TST·    R3                      ;IMMEDIATE·TRANSFER?·
002462· 003031                        BGT·    30$                     ;IF·NOT,·EXACTLY·ENOUGH·SPACE·LEFT·
002464  005203                        INC·    R3                      ;CHANGE·TYPE·OF·MOVE·
002466  010267  001750                MOV·    R2,UHLSAV·              ;FROM·IMMEDIATE·TRANSFER·
002472· 012702· 004442'               MOV·    #UHLSAV,R2·             ;TO·INDIRECT·TRANSFER·
002476  000752·                       BR      HPUT                    ;REST·BUFFER·SPACE·LEFT·
002500                          20$:
002500  016777  002012  001742·        MOV·    EOBFLG,@HRLOUT          ;SET·END·OF·BUFFER·SENTINEL·
002506  012701  004570'                MOV·    #IOCB3,R1               ;SET·UP·FOR·HRL·WRITE·
002512·                               CALL·   RELBUF·                 ;OUTPUT·CURRENT·BUFFER·
002516  016705  001702                MOV·    BSTADR,R5               ;GET·BATCH·STATUS·TABLE·
002522· 005365  000120                DEC·    B.HBLK(R5)              ;REDUCE·AVAILABLE·HRL·FILE·BLOCK·COUNT·
002526                                CALL·   GETBUF·                 ;GET·NEW·BUFFER·
002532· 016167  000014  001710        MOV·    CBUF(R1),HRLOUT·        ;SET·BUFFER·POINTER·
002540  012767  002000  001710        MOV·    #N.BUFW,SCOUNT          ;SET·SPACE·LEFT·IN·BUFFER·
002546                          30$:
002546  160367  001704                SUB·    R3,SCOUNT·              ;ADJUST·SPACE·LEFT·IN·BUFFER·
002552· 016704  001672                MOV·    HRLOUT,R4               ;SET·DESTINATION·ADDRESS·
002556                          35$:
002556  012224                        MOV·    (R2)+,(R4)+             ;MOVE·WORD·OF·ENTRY/AUTOINCREMENT·
002560  077302·                       SOB·    R3,35$                  ;NEXT·WORD·OF·ENTRY·
002562· 010467  001662                MOV·    R4,HRLOUT·              ;ADJUST·BUFFER·POINTER·
002566                          40$:
002566                          RESTOR·
002566  000207                  RETURN·
```

```
                                                .SBTTL· BUFFER·CONTROL ROUTINES·
                                        ;
                                        ;       GET/RELEASE·BUFFER·
                                        ;       THESE·SUBROUTINES·ARE·REENTRANT·AND·REQUIRE·THAT·R1 -> DESIRED IOCB·
                                        ;
002570                                  GETBUF:
002570                                          SAVE·   R2·
002572· 116102· 000016                          MOVB·   NXTG(R1),R2·            ;GET·OFFSET·TO·INTENDED·BUFFER
002576  032762· 000001  005114'                  BIT·    #BUFLOK,BDB(R2)         ;IS·NEXTG·BUFFER·ASSIGNABLE·YET·
002604                                          BOFF·   20$                     ;BRANCH·IF·IT·IS·
002606  105761  000021                          TSTB·   NXTR+1(R1)              ;HAS·I/O·BEEN·STARTED·ON·THIS·IOCB?
002612· 001002·                                 BNE·    10$                     ;IF·IT·HAS, WE·ARE·I/O·BOUND·- WAIT·
002614                                          CALL·   BUFIO·                  ;IF·NOT, INPUT·BUFFERS·NEED·TO BE·PRIMED
002620                                  10$:
002620                                          WAIT$   FDBAD(R1),EVNT(R1)      ;WAIT·FOR·I/O·COMPLETION·
002636                                  20$:
002636  016261  005114' 000014                  MOV·    BDB(R2),CBUF(R1)        ;LOAD·ADDRESS·OF·ASSIGNED·BUFFER·
002644  000423                                  BR      BUFXIT·
002646                                  ;
002646                                  RELBUF:
002646                                          SAVE·   R2·
002650  116102· 000016                          MOVB·   NXTG(R1),R2·            ;GET·OFFSET·TO·INTENDED·BUFFER
002654  052762· 000001  005114'                  BIS·    #BUFLOK,BDB(R2)         ;MARK·BUFFER·UNASSIGNABLE·
002662·                                         CALL·   BUFIO·                  ;START·UP·I/O·IF·POSSIBLE·
002666  126161  000016  000023                  CMPB·   NXTG(R1),LBDB(R1)       ;LAST·BUFFER·FOR·THIS·FILE?·
002674  002004                                  BGE·    15$                     ;RECYCLE·IF·YES·
002676  062761  000002  000016                  ADD·    #2,NXTG(R1)             ;SELECT·NEXT·BUFFER·TO·BE·ASSIGNED·
002704  000403                                  BR      BUFXIT·
002706                                  15$:
002706  116161  000022  000016                  MOVB·   IBDB(R1),NXTG(R1)       ;SELECT·FIRST·BUFFER
002714                                  BUFXIT:
002714                                          RESTOR· R2·
002716  000207                                  RETURN·
```

```
                                        ;
                                        ;       ASYNCHRONOUS EXIT ROUTINE
                                        ;
                                        ASTX:
                                                MOV     R1,R1SAV                ;SAVE R1
                                                MOV     (SP),R1                 ;GET IOST (SAME AS IOCB) ADDRESS FROM STACK
                                                MOVB    NXTR(R1),R1             ;GET OFFSET TO I/O ACTIVE BUFFER
                                                BIC     #BUFLOK,BDB(R1)         ;UNLOCK THE BUFFER
                                                MOV     (SP)+,R1                ;POP IOCB ADDRESS FROM STACK
                                                INC     CBLK(R1)                ;POINT TO NEXT LOGICAL BLOCK
                                                CLRB    NXTR+1(R1)              ;RESET I/O IN PROGRESS FLAG
                                                CMPB    NXTR(R1),LBDB(R1)       ;WAS THIS THE LAST BUFFER FOR FILE?
                                                BGE     10$                     ;RECYCLE IF YES
                                                ADD     #2,NXTR(R1)             ;SELECT NEXT BUFFER FOR I/O
                                                BR      20$
                                        10$:
                                                MOVB    IBDB(R1),NXTR(R1)       ;NEXT I/O BUFFER IS FIRST
                                        ;
                                        ;       START I/O ON NEXT BUFFER
                                        20$:
                                                SAVE    R0
                                                MOV     FDBAD(R1),R0            ;GET FDB OF COMPLETED I/O
                                                SAVE    F.BKDS+2(R0),F.BKST(R0) ;PRESERVE ASTX ROUTINE REENTRANCY
                                                SAVE    F.BKVB+2(R0),F.EFN(R0)
                                                CALL    BUFIO
                                                RESTOR  F.BKVB+2(R0),F.EFN(R0)
                                                RESTOR  F.BKDS+2(R0),F.BKST(R0)
                                                RESTOR  R0
                                                MOV     R1SAV,R1
                                                ASTX$S
                                        ;
                                        ;
                                        ;       CALCULATE LAST PHYSICAL BLOCK OF FILE
                                        ;
                                        GTBLK:
                                                MOV     FDBAD(R1),R0            ;GET FILE'S FDB
                                                MOV     F.HIBK+2(R0),R3         ;GET LAST ALLOCATED BLOCK
                                                ASR     R3                      ;CONVERT TO LOGICAL BLOCKS
                                                ASR     R3                      ;BY DIVIDING BY 4
                                                RETURN
                                        ;
                                        ;
                                        ;       BUFFER CONTROL ROUTINE SYMBOLIC VARIABLES
                                        ;
                                        BUFLOK  =       BIT0                    ;BUFFER LOCKED, CANNOT BE ASSIGNED
                                        BUFRD   =       BIT2                    ;READ OPERATIONS ON THIS IOCB
                                        BUFWR   =       BIT3                    ;WRITE OPERATIONS ON THIS IOCB
                                        BUFRAP  =       BIT4                    ;FILE WRAP AROUND IN EFFECT
                                        BUFEOF  =       BIT5                    ;LAST LOGICAL I/O BLOCK ENCOUNTERED
                                        BUFOPN  =       BIT7                    ;IOCB HAS BEEN INITIALIZED
```

```
                                        ;       PERFORM·I/O·ON·NEXT·AVAILABLE·BUFFER.
                                        ;
                                        ;       R1 -> BUFFER DEFINITION·BLOCK.
                                        ;
003104                                  BUFIO:
003104
003114  132761  000200  000007                BITB·   #BUFOPN,ATTR(R1)      ;HAS·IOCB·BEEN·INITIALIZED·YET?·
003122·                                         BON·    BUFQIO·               ;IF·SO, SET·UP·I/O·OPERATION·
003124  005761  000010                          TST·    CBLK(R1)              ;HAS·LOGICAL·START·OF·FILE·BEEN·SPECIFIED?·
003130  003003                                  BGT·    10$                   ;BRANCH·IF·IT·HAS·
003132· 012761  000001  000010                  MOV·    #1,CBLK(R1)           ;SET·START·AT·BEGINNING·OF·FILE·
003140                                  10$:
003140  005761  000012                          TST·    LBLK(R1)              ;HAS·LOGICAL·END·BEEN·SPECIFIED?·
003144  003010                                  BGT·    20$                   ;BRANCH·IF·IT·HAS·
003146  132761  000020  000007                BITB·   #BUFRAP,ATTR(R1)      ;HAS·WRAP·AROUND·BEEN·SPECIFIED?·
003154                                          BON·    20$                   ;BRANCH·IF·IT·HAS·
003156                                          CALL·   GTBLK·                ;CALCULATE·LOGICAL·BLOCK·SIZE·OF·FILE·
003162· 010361  000012                          MOV·    R3,LBLK(R1)           ;SET·UP·LOGICAL·BLOCK·TO·STOP·I/O·
003166                                  20$:
003166  132761  000004  000007                BITB·   #BUFRD,ATTR(R1)       ;IS·THIS·IOCB·FOR·READ·OPERATIONS?·
003174                                          BOFF·   30$                   ;BRANCH·IF·NOT·
003176  116102· 000022                          MOVB·   IBDB(R1),R2·          ;GET·OFFSET·TO·FIRST·BUFFER·
003202·                                 25$:
003202· 052762· 000001  005114'                 BIS·    #BUFLOK,BDB(R2)       ;MARK·BUFFER·UNASSIGNABLE·
003210  062702· 000002                          ADD·    #2,R2·                ;STEP·TO·NEXT·BUFFER
003214  120261  000023                          CMPB·   R2,LBDB(R1)           ;LAST·BUFFER·BEEN·MARKED?·
003220  003770                                  BLE·    25$                   ;IF·NOT, GO·TO·MARK·IT·
003222·                                 30$:
003222·                                         FDBK$R· FDBAD(R1),,,,,,#ASTX·  ;DEFINE·AST·EXIT·
003234  152761  000200  000007                BISB·   #BUFOPN,ATTR(R1)      ;SET·IOCB·INITIALIZED·FLAG·
                                        ;
                                        ;       SET·UP·I/O·OPERATION.
                                        ;
003242                                  BUFQIO:
003242· 132761  000040  000007                BITB·   #BUFEOF,ATTR(R1)      ;HAS·FILE'S·LOGICAL·EOF·ALREADY·BEEN·REACHED?·
003250                                          BON·    BUFIOX·               ;IF·SO, EXIT·
003252· 105761  000021                          TSTB·   NXTR+1(R1)            ;IS·I/O·ALREADY·IN·PROGRESS·ON·THIS·IOCB?·
003256  001152·                                 BNE·    BUFIOX·               ;EXIT·IF·SO·
003260  016102· 0000,'0                        MOV·    NXTR(R1),R2·          ;GET·OFFSET·TO·NEXT·BUFFER·FOR·I/O·
003264  032762· 000001  005114'                 BIT·    #BUFLOK,BDB(R2)       ;IS·BUFFER·AVAILABLE·FOR·I/O?·
003272·                                         BOFF·   BUFIOX·               ;BRANCH·IF·NOT·
003274  016203  005114'                         MOV·    BDB(R2),R3            ;GET·BUFFER·ADDRESS·
003300  042703  000001                          BIC·    #BUFLOK,R3            ;CLEAR·LOCK·FLAG·
003304                                          FDBK$R· FDBAD(R1),R3          ;DEFINE·BUFFER·IN·THE·FDB·
003314                                          FDBK$R· ,,,,EVNT(R1),R1       ;DEFINE·EVENT·FLAG·AND·IOST·BLOCK·
```

```
                                        ;       DO THE APPROPRIATE I/O OPERATION
                                        ;
003326                                  10$:
003326  016103  000012                          MOV     LBLK(R1),R3             ;GET LAST LOGICAL BLOCK
003332  105261  000017                          INCB    NXTG+1(R1)              ;SET LOGICAL BLOCK INDICATOR
003336                                  20$:
003336  026103  000010                          CMP     CBLK(R1),R3             ;COMPARE CURRENT WITH LAST
003342  002427                                  BLT     50$                     ;DO I/O - NOT LAST BLOCK
003344  001420                                  BEQ     40$                     ;DO I/O - TEST FOR LAST BLOCK
                                        ;
003346  132761  000020  000007                  BITB    #BUFRAP,ATTR(R1)        ;LAST HIGH - FILE WRAP AROUND?
003354                                          BOFF    BUFIOX                  ;HAVE ALREADY REACHED END OF FILE
003356  105761  000017                          TSTB    NXTG+1(R1)              ;LOGICAL BLOCK INDICATOR SET?
003362  001004                                  BNE     30$                     ;BRANCH IF YES
003364  012761  000001  000010                  MOV     #1,CBLK(R1)             ;WRAP AROUND TO START OF FILE
003372  000755                                  BR      10$                     ;RETEST FOR LOGICAL EOF
003374                                  30$:
003374                                          CALL    GTBLK                   ;CALCULATE LAST PHYSICAL BLOCK IN FILE
003400  105061  000017                          CLRB    NXTG+1(R1)              ;RESET LOGICAL BLOCK INDICATOR
003404  000754                                  BR      20$                     ;RETEST FOR PHYSICAL END OF FILE
003406                                  40$:
003406  105761  000017                          TSTB    NXTG+1(R1)              ;WHICH END OF FILE IS BEING TESTED?
003412  001405                                  BEQ     60$                     ;BRANCH IF PHYSICAL EOF
003414  152761  000040  000007                  BISB    #BUFEOF,ATTR(R1)        ;SET LOGICAL EOF REACHED INDICATOR
003422                                  50$:
003422  105061  000017                          CLRB    NXTG+1(R1)              ;RESET LOGICAL BLOCK INDICATOR
                                        ;
                                        ;       CALCULATE CURRENT VIRTUAL BLOCK FROM CBLK FIELD
                                        ;
003426                                  60$:
003426  016103  000010                          MOV     CBLK(R1),R3             ;GET CURRENT LOGICAL BLOCK
003432  005303                                  DEC     R3
003434  006303                                  ASL     R3
003436  006303                                  ASL     R3
003440  005203                                  INC     R3                      ;VB = (CBLK-1)*4+1
003442  016100  000004                          MOV     FDBAD(R1),R0            ;GET FDB ADDRESS
003446  010360  000066                          MOV     R3,F.BKVB+2(R0)         ;INITIALIZE VIRTUAL BLOCK ADDRESS IN FDB
003452  105261  000021                          INCB    NXTR+1(R1)              ;SET I/O IN PROGRESS
003456  132761  000004  000007                  BITB    #BUFRD,ATTR(R1)         ;READ SPECIFIED?
003464                                          BOFF    70$                     ;BRANCH IF NOT
003466                                          READ$                           ;PERFORM THE READ
003472  000406                                  BR      80$                     ;CHECK SUCCESS
003474                                  70$:
003474  132761  000010  000007                  BITB    #BUFWR,ATTR(R1)         ;WAS WRITE SPECIFIED?
003502                                          BOFF    BUFIOX                  ;BRANCH IF NOT
003504                                          WRITE$                          ;PERFORM THE WRITE
003510                                  80$:
003510  103035                                  BCC     BUFIOX                  ;EXIT IF NO ERROR
```

```
                                        ;       ERROR·EXIT·ON·FCS·DIRECTIVE·
                                        ;
003512·                                 ERRX:
003512·  126027  000052  000000G·               CMPB·   F.ERR(R0),#IE.EOF·      ;WAS·THIS·READ·PAST·LAST·BLOCK
003520   001006                                 BNE·    10$                     ;IF·NOT, A TRUE·ERROR·
003522·  152761  000040  000007                 BISB·   #BUFEOF,ATTR(R1)        ;SET·END·OF·FILE·FLAG·
003530   105061  000021                         CLRB·   NXTR+1(R1)              ;RESET·I/O IN·PROGRESS·FLAG·
003534   000423                                 BR      BUFIOX·                 ;EXIT·ROUTINE·
003536                                  10$:
003536·  010167  000636                         MOV·    R1,PAR1
003542·  116067  000052  000632·                MOVB·   F.ERR(R0),PAR2
003550   152767  000377  000625                 BISB·   #255.,PAR2+1
003556                                          MOUT$S· #MSG6,#MSGPAR·
003576                                          EXIT$S·
                                        ;
                                        ;       RETURN·POINT FOR·BUFIO SUBROUTINE·
                                        ;
003604                                  BUFIOX:
003604                                          RESTOR· R0,R2,R3,R4
003614·  000207                                 RETURN·
```

```
                                                        .SBTTL· ERROR·HANDLING ROUTINE·
                                                        .NLIST· BEX·
                                                ;
                                                ;   DIRECTIVE·ERROR·
                                                ;
003616  011667  000556                          DIRERR: MOV·     (SP),PAR1
003622· 116700  000000G·                                MOVB·    $DSW,R0
003626  010067  000550                                  MOV·     R0,PAR2·
003632·                                                 MOUT$S·  #MSGD,#PAR1
003652· 000207                                          RETURN·
                                                ;
003654                                                  .PSECT·
                                                ;
                                                ;   STRING DESCRIPTORS·
                                                ;
003654  000044                                  MSG0:   .WORD·   LN0E-LN0
003656  003714'                                         .WORD·   LN0
003660  000034                                  MSG1:   .WORD·   LN1E-LN1
003662· 003760'                                         .WORD·   LN1
003664  000041                                  MSG2:   .WORD·   LN2E-LN2·
003666  004014'                                         .WORD·   LN2·
003670  000050                                  MSG3:   .WORD·   LN3E-LN3
003672· 004055'                                         .WORD·   LN3
003674  000062·                                 MSG4:   .WORD·   LN4E-LN4
003676  004125'                                         .WORD·   LN4
003700  000034                                  MSG5:   .WORD·   LN5E-LN5
003702· 004207'                                         .WORD·   LN5
003704  000062·                                 MSG6:   .WORD·   LN6E-LN6
003706  004243'                                         .WORD·   LN6
003710  000040                                  MSGD:   .WORD·   LNDE-LND·
003712· 004325'                                         .WORD·   LND·
                                                ;
                                                ;   FORMAT STRINGS·
                                                ;
003714      110     122     114                 LN0:    .ASCIZ· /HRLMRG·PROGRAM·EXIT·- ACK·CODE·=·%D/·
003760                                          LN0E:
003760      110     122     114                 LN1:    .ASCIZ· /HRL MERGE·FILE·OPEN·FAILURE/·
004014                                          LN1E:
004014      125     110     114                 LN2:    .ASCIZ· /UHL OPEN·FAILURE, QUERY ID·=·%1D/
004055                                          LN2E:
004055      111     116     126                 LN3:    .ASCIZ· /INVALID·COMMAND·FROM·MSCHED, CODE =·%1O/·
004125                                          LN3E:
004125      111     116     103                 LN4:    .ASCIZ· /INCONSISTENT·BATCH·IDS, OPEN·=·%1D, COMMAND =·%1D/·
004207                                          LN4E:
004207      111     116     126                 LN5:    .ASCIZ· /INVALID·QUERY ID, QID·= %1O/·
004243                                          LN5E:
004243      102·    125     106                 LN6:    .ASCIZ· /BUFIO·FAILURE, IOCB·ADDRESS·=·%O, ERROR·CODE·=·%O/·
004325                                          LN6E:
004325      120     103     040                 LND:    .ASCIZ· /PC·=·%1O, DIRECTIVE·ERROR·=·%1D/·
004365                                          LNDE:
                                                        .EVEN·
```

```
                                        .SBTTL  DATA STORAGE AND SYMBOLIC VARIABLES
                                ;
                                        .PSECT  IOCBOF,ABS
000000                  IOST:   .BLKW   2               ;I/O STATUS BLOCK
000000                  FDBAD:  .BLKW   1               ;ADDRESS OF FDB
000004                  EVNT:   .BLKB   1               ;ADDRESS OF EVENT FLAG
000006                  ATTR:   .BLKB   1               ;TYPE OF I/O
000007                  CBLK:   .BLKW   1               ;CURRENT LOGICAL BLOCK
000010                  LBLK:   .BLKW   1               ;LAST BLOCK (IF WRAP AROUND)
000012                  CBUF:   .BLKW   1               ;ADDRESS OF BUFFER CURRENTLY IN USE
000014                  NXTG:   .BLKW   1               ;BDB OFFSET OF NEXT ENTRY FOR ASSIGNMENT
000016                  NXTR:   .BLKW   1               ;BDB OFFSET OF NEXT ENTRY FOR I/O
000020                  IBDB:   .BLKB   1               ;FILE'S FIRST BDB INDEX
000022                  LBDB:   .BLKB   1               ;FILE'S LAST BDB INDEX
000023                          .PSECT
004366
                                ;
004366                  RCVB:   .BLKW   2               ;RECEIVE COMMAND BLOCK
004372                  PGMID:  .BLKB   1               ;PROGRAM ID
004373                  CMDX:   .BLKB   1               ;COMMAND/ACK CODE
004374                  BCHNO:  .BLKW   1               ;BATCH NUMBER
004376                  QRYID:  .BLKW   1               ;QUERY ID
                                ;
004400                  MSGPAR: .BLKW   0               ;MSGOUT PARMS/RCVB BLOCK PAD
004400  000000          PAR1:   .WORD   0
004402  000000          PAR2:   .WORD   0
004404  000000          PAR3:   .WORD   0
004406                          .BLKW   7
004424  000000          BSTADR: .WORD   0               ;BATCH STATUS TABLE ADDRESS
004426  000000          NBUHL:  .WORD   0               ;NUMBER OF UHL SUBFILE LOGICAL BLOCKS
004430  000000          STUHL:  .WORD   0               ;LOGICAL BLOCK START OF UHL SUBFILE
004432  000000          FDUHL:  .WORD   0               ;FDSC ADDRESS OF QUERY SPOOL FILE
004434  000000          EOHRL:  .WORD   0               ;END OF HRL FLAG
004436  000000          UHLCNT: .WORD   0               ;REMAINING NUMBER OF UHL ENTRIES
004440  000000          UHLNXT: .WORD   0               ;UHL INPUT BUFFER POINTER
004442  000000          UHLSAV: .WORD   0               ;IMMEDIATE SEGMENT TRANSFER BUFFER
004444  000000          UHLWD:  .WORD   0               ;SPACE LEFT IN UHL BUFFER
004446  000000          HRLNXT: .WORD   0               ;INPUT HRL BUFFER POINTER
004450  000000          HRLOUT: .WORD   0               ;OUTPUT HRL BUFFER POINTER
004452  000000          HRLCNT: .WORD   0               ;INPUT HRL BUFFER SPACE
004454  000000          QCOUNT: .WORD   0               ;NUMBER OF QIDS IN HRL ENTRY
004456  000000          SCOUNT: .WORD   0               ;SPACE LEFT IN HRL OUTPUT BUFFER
004460  000000          RISAV:  .WORD   0               ;ASTX SAVE AREA
004462  000000          DCNT:   .WORD   0               ;PARTIAL DOC ID WORD COUNTER
004464  000000          ABRTFL: .WORD   0               ;HRL ABORT COMMAND FLAG
004466                  ;
004466                  UDOC:   .BLKW   3               ;UHL ENTRY BUFFER + SKELETON HRL WORD
004474  000001                  .WORD   1               ;SINGLE QUERY ID
004476                  HDOC:   .BLKW   1               ;INPUT HRL ENTRY POINTER
004500  000017          EOFDOC: .WORD   15.             ;DUMMY DOC ID
004502  177777                  .WORD   -1
004504  177777                  .WORD   -1
                                ;
004506  052510          UHLHDR: .WORD   "HU             ;UHL SUBFILE HEADER
004510  041461          HRLHDR: .WORD   "1C             ;HRL FILE HEADER RECORD
004512  000000                  .WORD   0               ;PAD WORD
004514  140000          EODFLG: .WORD   BIT15!BIT14     ;END OF DATA SENTINEL
004516  040000          EOBFLG: .WORD   BIT14           ;END OF BUFFER SENTINEL
```

```
                        .SBTTL  FILE STRUCTURES
                ;
                ;       I/O CONTROL BLOCK - UHL FILE
                ;
                IOCB1:
934
935
936
937
938 004520              .BLKW   2               ;IOST BLOCK
939 004520              .WORD   FDB1            ;FDB ADDRESS
940 004524  004614'     .BYTE   1               ;EVENT FLAG 1
941 004526      001     .BYTE   BUFRD           ;READ/NO WRAP AROUND
942 004527      004     .BLKW   1               ;CURRENT LOGICAL BLOCK
943 004530              .BLKW   1               ;LAST LOGICAL BLOCK
944 004532              .WORD   0               ;ADDRESS OF BUFFER IN USE
945 004534  000000      .BYTE   BDB1-BDB        ;INDEX TO INITIAL ASSIGN BDB
946 004536      000     .BYTE   0               ;INITIALLY UNASSIGNABLE
947 004537      000     .BYTE   BDB1-BDB        ;INDEX TO INITIAL I/O BDB
948 004540      000     .BYTE   0               ;INITIALLY UNASSIGNABLE
949 004541      000     .BYTE   BDB1-BDB        ;INDEX TO FILE'S FIRST BDB
950 004542      000     .BYTE   BDB2-BDB        ;INDEX TO FILE'S LAST BDB
951 004543      002
952
                ;
                ;       I/O CONTROL BLOCK - HRL FILE INPUT
                ;
                IOCB2:
953
954
955 004544              .BLKW   2               ;IOST BLOCK
956 004544              .WORD   FDB2            ;FDB ADDRESS
957 004550  004754'     .BYTE   2               ;EVENT FLAG 2
958 004552      002     .BYTE   BUFRD!BUFRAP    ;READ WITH WRAP AROUND
959 004553      024     .BLKW   1               ;CURRENT LOGICAL BLOCK
960 004554              .BLKW   1               ;LAST LOGICAL BLOCK
961 004556              .WORD   0               ;ADDRESS OF BUFFER IN USE
962 004560  000000      .WORD   BDB3-BDB        ;INDEX TO INITIAL ASSIGN BDB
963 004562  000004      .WORD   BDB3-BDB        ;INDEX TO INITIAL I/O BDB
964 004564  000004      .BYTE   BDB3-BDB        ;INDEX TO FILE'S FIRST BDB
965 004566      004     .BYTE   BDB4-BDB        ;INDEX TO FILE'S LAST BDB
966 004567      006
967
                ;
                ;       I/O CONTROL BLOCK BLOCK - HRL FILE OUTPUT
                ;
968
969
                IOCB3:
970 004570
971 004570              .BLKW   2               ;IOST BLOCK
972 004574  004754'     .WORD   FDB2            ;FDB ADDRESS
973 004576      003     .BYTE   3               ;EVENT FLAG 3
974 004577      030     .BYTE   BUFWR!BUFRAP    ;WRITE WITH WRAP AROUND
975 004600              .BLKW   1               ;CURRENT LOGICAL BLOCK
976 004602              .BLKW   1               ;LAST LOGICAL BLOCK
977 004604  000000      .WORD   0               ;ADDRESS OF BUFFER IN USE
978 004606  000010      .WORD   BDB5-BDB        ;INDEX TO INITIAL ASSIGN BDB
979 004610  000010      .WORD   BDB5-BDB        ;INDEX TO INITIAL I/O BDB
980 004612      010     .BYTE   BDB5-BDB        ;INDEX TO FILE'S FIRST BDB
981 004613      012     .BYTE   BDB6-BDB        ;INDEX TO FILE'S LAST BDB
```

```
                                        ;       FILE·DEFINITION·BLOCKS
                                        ;
                                        ;       UHL·FILE·
                                        ;
004614                                  FDB1:   FDBDF$
004754                                          FDRC$A·  FD,RWM·
004754                                          FDBK$A·  ,2*N.BUFW,,1,IOCB1
004754                                          FDOP$A·  1
                                        ;
                                        ;       HRL·FILE·
                                        ;
004754                                  FDB2:   FDBDF$
005114                                          FDRC$A·  FD,RWM·
005114                                          FDBK$A·  ,2*N.BUFW,,2,IOCB2·
005114                                          FDOP$A·  2·
                                        ;
005114                                          FSRSZ$   0
                                        ;
                                        ;       BUFFER·DEFINITION·BLOCK·ADDRESS·VECTOR·
                                        ;
005114                                  BDB:
005114  005130'                         BDB1:   .WORD·   BUF1
005116  011130'                         BDB2:   .WORD·   BUF2
005120  015130'                         BDB3:   .WORD·   BUF3
005122· 021130'                         BDB4:   .WORD·   BUF4
005124  025130'                         BDB5:   .WORD·   BUF5
005126  031130'                         BDB6:   .WORD·   BUF6
                                        ;
                                        ;       I/O·BUFFERS·
                                        ;
005130                                  BUF1:   .BLKW·   N.BUFW·
011130                                  BUF2:   .BLKW·   N.BUFW·
015130                                  BUF3:   .BLKW·   N.BUFW·
021130                                  BUF4:   .BLKW·   N.BUFW·
025130                                  BUF5:   .BLKW·   N.BUFW·
031130                                  BUF6:   .BLKW·   N.BUFW·
                                        ;
          000000'                               .END·    HRLMRG·
```

SYMBOL TABLE


```
ABRTFL  004464R
ASTX    002720R
ATTR    000007      015
BCHNO   004374R
BDB     005114R
BDB1    005114R
BDB2    005116R
BDB3    005120R
BDB4    005122R
BDB5    005124R
BDB6    005126R
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BLDEFL= ****** GX
BLDNFL= ****** GX
BSTADR  004424R
BSTPTR= ****** GX
BS.CLS= 000002
BS.DBU= 000004
BS.INA= 000000
BS.OPN= 000001
BS.SRC= 000003
BUFEOF= 000040
BUFIO   003104R
BUFIOX  003604R
BUFLOK= 000001
BUFOPN= 000200
BUFQIO  003242R
BUFRAP= 000020
BUFRD = 000004
BUFWR = 000010
BUFXIT  002714R
BUF1    005130R
BUF2    011130R
BUF3    015130R
BUF4    021130R
BUF5    025130R
BUF6    031130R
BYTE0 = 000000
BYTE1 = 000001
BYTE2 = 000002
BYTE3 = 000003
BYTE4 = 000004
BYTE5 = 000005
BYTE6 = 000006      015
BYTE7 = 000007
BYTE8 = 000010
BYTE9 = 000011
BYTVAL= 000012
B.BSTA  000054
B.CNTX  000046
B.CQUQ  000060
B.FEMA  000132
B.FEMB  000142
B.FEMC  000152
B.FFSA  000202
B.FFSB  000212
B.FFSC  000222
B.FMHR  000172
B.FQLS  000162
B.FSAZ  000100
B.FSBZ  000102
B.FSCZ  000104
B.HBLK  000120
B.HDOC  000114
B.HRLP  000126
B.HRLR  000122
B.HRLW  000124
B.NMBR  000052
B.NQRY  000232
B.QLSZ  000106
B.QMAP  000234
B.QSPL  000316
B.QTTM  000076
B.QUQP  000056
B.SFDB  000010
B.SIZE  000772
B.SNDP  000012
B.SSQ   000004
B.SSQF  000050
B.STAT  000044
B.STTE  000053
B.UDOC  000110
CBLK    000010
CBUF    000014
CF.B0 = 000070
CF.B2 = 000067
CF.B4 = 000066
CF.B6 = 000065
CF.DR0= 000064
CF.DR1= 000063
CMDLST  000114R
CMDTAB  000104R
CMDX    004373R
DBSLEN= 000116
DCNT    004462R
DH.BF0  000002
DH.BF1  000004
DH.CTL  000000
DH.DMC  000010
DH.FLG  000006
DIRERR  003616R
DN.DCK  000000
DN.NTP  000004
DN.NXT  000006
DN.ROT  000002
010 DN.SIZ  000010
010 EOBFLG  004516R
010 EODFLG  004514R
010 EOFDOC  004500R
010 EOHRL   004434R
010 ERRX    003512R
010 EVNT    000006
010 FDBAD   000004
010 FDB1    004614R
010 FDB2    004754R
010 FDUHL   004432R
010 FD.FID  000000
010 FD.FNB  000006
010 FD.FVR  000004
010 FD.LEN  000010
010 FD.RWM= ****** GX
010 FN.DBR  000026
010 FN.DBS  000022
010 FN.DHR  000040
010 FN.EMA  000012
010 FN.EMB  000014
010 FN.EMC  000016
010 FN.FSA  000000
010 FN.FSB  000002
010 FN.FSC  000004
010 FN.LGQ  000034
010 FN.LGU  000036
010 FN.MFD  000024
010 FN.MHR  000010
010 FN.NMB  000044
010 FN.QLS  000006
010 FN.QRY  000020
010 FN.SF0  000030
010 FN.SF1  000032
015 FN.SHD  000042
015 FO.MFY= ****** GX
    FO.RD = ****** GX
    FO.WRT= ****** GX
    F.ACTL= 000076
    F.ALOC= 000040
    F.BBFS= 000062
    F.BDB = 000070
    F.BGBC= 000057
    F.BKDN= 000026
    F.BKDS= 000020
    F.BKEF= 000050
    F.BKP1= 000051
005 F.BKST= 000024
005 F.BKVB= 000064
005 F.CHR = 000075
005 F.CNTG= 000034
005 F.DFNB= 000046
    F.DSPT= 000044
013 F.DVNM= 000134
013 F.EFBK= 000010
013 F.EFN = 000050
013 F.EOBB= 000032
013 F.ERR = 000052
    F.FACC= 000043
    F.FFBY= 000014
    F.FNAM= 000110
    F.FNB = 000102
    F.FTYP= 000116
015 F.FVER= 000120
015 F.HIBK= 000004
    F.LUN = 000042
    F.MBCT= 000054
    F.MBC1= 000055
003 F.MBFG= 000056
003 F.NRBD= 000024
003 F.NREC= 000030
003 F.OVBS= 000030
    F.RACC= 000016
011 F.RATT= 000001
011 F.RCNM= 000034
011 F.RCTL= 000017
011 F.RSIZ= 000002
011 F.RTYP= 000000
011 F.SEQN= 000100
011 F.SPDV= 000072
011 F.SPUN= 000074
011 F.STBK= 000036
011 F.UNIT= 000136
011 F.URBD= 000020
011 F.VBN = 000064
011 F.VBSZ= 000060
011 GETBUF  002570R
011 GETCMD  000004R
011 GETFRE= ****** GX
011 GTBLK   003066R
011 HDOC    004476R
011 HGET    002100R
    HGETX   002244R
    HLABRT  000542R
    HLINIT  001342R
    HLMERG  000164R
    HLMERX  000436R
    HLNITX  001756R
    HLPRBX  001212R
    HLPROB  000724R
    HLTERM  000626R
    HPUT    002424R
    HRLCNT  004452R
    HRLHDR  004510R
    HRLMRG  000000R
    HRLNTH  000612R
    HRLNXT  004446R
    HRLOUT  004450R
    IBDB    000022      015
    IE.EOF= ****** GX
    IOCB1   004520R
    IOCB2   004544R
    IOCB3   004570R
    IOST    000000      015
    LBDB    000023      015
    LBLK    000012      015
    LND     004325R
    LNDE    004365R
    LN0     003714R
    LN0E    003760R
    LN1     003760R
    LN1E    004014R
    LN2     004014R
    LN2E    004055R
    LN3     004055R
    LN3E    004125R
    LN4     004125R
    LN4E    004207R
    LN5     004207R
    LN5E    004243R
    LN6     004243R
    LN6E    004325R
    LOCQRY  001214R
    LOCXIT  001336R
    M     = 000062
    MRGDEQ  000440R
    MRGDGT  000574R
    MRGDLT  000550R
    MSGD    003710R
    MSGOUT= ****** GX
    MSGPAR  004400R
    MSG0    003654R
    MSG1    003660R
    MSG2    003664R
    MSG3    003670R
    MSG4    003674R
    MSG5    003700R
    MSG6    003704R
    N     = 000002
    NBUHL   004426R
    NXTG    000016      015
    NXTR    000020      015
    N.BFAC= 000004
    N.BHGH= 000006
    N.BTCH= 000004
    N.BUFB= 004000
    N.BUFW= 002000
    N.DID = 000024
    N.DVNM= 000032
    N.FID = 000000
    N.FNAM= 000006
    N.FOBA= 000064
    N.FTYP= 000014
    N.FVER= 000016
```

```
HRLMRG  M[illegible]0.M1110  27-MAR-80 13:29  PAGE 35-2
SYMBOL TABLE
```

```
N.NEXT= 000022          SR.DAY  000010      002 SS.STT  000000      004 S.HRL = 000240          XBATCH= 000013
N.PKSZ= 000020          SR.DLT  000014      002 STUHL   004430R         S.NFEN= 000020          XDBLOA= 000004
N.PKTS= 000043          SR.ECB  000047      002 ST.ASZ  000020      006 UDOC    004466R         XDBPRO= 000012
N.QURY= 000031          SR.ECH  000046      002 ST.BSZ  000024      006 UGET    002246R         XDMCIN= 000006
N.STAT= 000020          SR.ECL  000050      002 ST.BTC  000000      006 UHLCNT  004436R         XFOSMR= 000007
N.SUNT= 000002          SR.FIB  000012      002 ST.CSZ  000030      006 UHLHDR  004506R         XGTSRE= 000014
N.UNIT= 000034          SR.GRE  000100      002 ST.HRL  000010      006 UHLNXT  004440R         XHITSK= 000011
PAR$$$= 000061          SR.GRS  000072      002 ST.LEN  000044      006 UHLSAV  004442R         XHLMER= 000002
PAR1    004400R         SR.LEN  000122      002 ST.QRY  000002      006 UHLWD   004444R         XHOTSK= 000010
PAR2    004402R         SR.LIN  000066      002 ST.QSZ  000034      006 ULINIT  001762R         XMSCHE= 000000
PAR3    004404R         SR.LIP  000062      002 ST.SCH  000040      006 ULNITX  002076R         XQTS  = 000003
PGMID   004372R         SR.MON  000006      002 ST.UHL  000004      006 WN.NTP  000004    012 XQT0  = 000001
PGMXIT  000156R         SR.NDC  000042      002 ST.XLT  000014      006 WN.NXT  000006    012 XSULOA= 000005
PUTSSQ= ****** GX       SR.NDS  000036      002 SU.DBU= 000004          WN.ROT  000002    012 $DSW  = ******  GX
QCOUNT  004454R         SR.NIN  000030      002 SU.DON= 000006          WN.SIZ  000010    012 $$$   = 000000R   014
QE.RO1= 000144          SR.NIP  000022      002 SU.IDL= 000000          WN.SRC  000000    012 $$$OST= 000010
QRYID   004376R         SR.SDB  000032      002 SU.LOD= 000001          WN.TYP  000001    012 $$$TI = 000000
Q.FDSC  000004      007 SR.SRC  000002      002 SU.SRC= 000002          WORD0 = 000000        .CLOSE= ****** G
Q.NQBK  000000      007 SR.SUN  000000      002 SU.SRR= 000005          WORD1 = 000002        .DLFNB= ****** GX
Q.NUHL  000002      007 SR.TWS  000056      002 SU.XPD= 000003          WORD2 = 000004        .FINIT= ****** G
Q.SIZE  000014      007 SR.WSL  000052      002 S.BFHD= 000020          WORD3 = 000006        .FSRCB= ****** G
RCVB    004366R         SR.YR   000004      002 S.FATT= 000016          WORD4 = 000010        .OPFNB= ****** G
RELBUF  002646R         SR.1IN  000024      002 S.FDB = 000140          WORD5 = 000012        .READ = ****** G
R.VXBA= 000006          SR.1IP  000016      002 S.FNAM= 000006          WORD6 = 000014        .WAIT = ****** G
R.VXTN= 000002          SS.FID  000002      004 S.FNB = 000036          WORD7 = 000016        .WRITE= ****** G
R1SAV   004460R         SS.FNB  000010      004 S.FNBW= 000017          WORD8 = 000020        ...PC1= 004754R
SCOUNT  004456R         SS.FVR  000006      004 S.FNTY= 000004          WORD9 = 000022        ...PC2= 005114R
SR.ARE  000114      002 SS.LEN  000012      004 S.FTYP= 000002          WRDVAL= 000024        ...TPC= 000020
SR.ARS  000106      002
```

```
. ABS.  000000  000
        035130  001
SRCOFF  000122  002
FDSCOF  000010  003
SUSOFF  000012  004
DHROFF  000012  005
STTOFF  000044  006
QSPLOF  000014  007
BSTOFF  000772  010
FNOFFS  000044  011
WNODOF  000010  012
DNODOF  000010  013
$DPB$$  000010  014
IOCBOF  000024  015
$$FSR1  000000  016
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  7132 WORDS  (28 PAGES)
DYNAMIC MEMORY:  8084 WORDS  ( 31 PAGES)
ELAPSED TIME:  00:01:14
HLMERG,HLMERG/-SP=[20,1]P,M,HLMERG
```

11- 20 FOSMRG·EXECUTABLE·CODE·
15- 174 ERROR·HANDLING ROUTINE·
16- 223 DATA STORAGE·FIELDS·AND·VARIABLES·
17- 239 FILE STRUCTURES·AND·RECORD·BUFFERS

```
                        .TITLE  FOSMRG PROGRAM
                ;
                ;   THE FOSMRG PROGRAM SUMS THE FLU OCCURRENCE COUNTERS RECEIVED FROM
                ;   EACH OF THE SEARCH UNITS.  THE INPUT FILES ARE ACCESSED SUCCESSIVELY,
                ;   ONE AT A TIME, THROUGH THE SEARCH STATUS TABLE.  THE FDSC OF THE
                ;   OUTPUT FILE IS QUEUED TO HOTSK AND ALSO, MSCHED IS ACKNOWLEDGED WITH
                ;   AN ACK1 IN THE SSQ.
                ;
                ;   FOSMRG CAN HANDLE AN ARBITRARY NUMBER OF SEARCH UNITS, SUBJECT TO
                ;   THESE TWO RESTRICTIONS: THE SEARCH UNIT STATUS ENTRIES ARE STORED
                ;   CONTIGUOUSLY STARTING AT BASE ADDRESS SUST (A GLOBAL SYMBOL).  ALSO,
                ;   THE NUMBER OF SEARCH UNITS IN THE SYSTEM IS DEFINED BY THE SYMBOL
                ;   N.SUNT.
                ;
                        .MCALL  EXIT$S,RCVX$C,SDAT$C,RSUM$C
                        .MCALL  FINIT$,FDBDF$,FDRC$A,FDBK$A,FDOP$A,FSRSZ$A
                        .MCALL  FDBK$R,FDAT$R,OFNB$R,OFNB$W,READ$,WRITE$,WAIT$,CLOSE$
000012                  .RADIX  10
```

```
                                        .SBTTL  FOSMRG EXECUTABLE CODE
000000                          FOSMRG:
                                        FINIT$
000004                          FOS.10:
                                        RCVX$C  ,RCVIN
000012  103004                         BCC     FOS.15
000014                                  CALL    DIRERR
000020  000167  000614                  JMP     FOS.90
000024                          FOS.15:
000024  012700  001300'                 MOV     #FDBIN,R0               ;FDB ADDRESS.
000030  016701  001230                  MOV     FDSCAD,R1               ;FDSC ADDRESS OF CURRENT S.U.
000034                                  CALL    BLDEFL                  ;BUILD FILE NAME BLOCK IN FDB
                                ;
                                ;       OPEN THE INPUT FILE - IF OPEN FAILS, REPORT NON-EXISTENCE OF FOS
                                ;       VIA MSGQUT AND GO ON TO THE NEXT INPUT FOS
                                ;
000040                                  OFNB$R  #FDBIN
000056  103003                         BCC     FOS.20                  ;IF OPEN OK, DO THE READ
000060                                  CALL    FCSERR                  ;REPORT NON-EXISTENT S.U. FOS
000064  000454                         BR      FOS.50                  ;SET UP NEXT INPUT FOS FILE
000066                          FOS.20:
000066                                  READ$   #FDBIN                  ;READ THE INPUT FOS FILE
000076                                  WAIT$   #FDBIN
000106  103003                         BCC     FOS.30                  ;IF OK, DO THE ACCUMULATION LOGIC
000110                                  CALL    FCSERR                  ;REPORT FOS READ ERROR
000114  000440                         BR      FOS.50                  ;SET UP NEXT INPUT FOS FILE
000116                          FOS.30:
000116  012700  001300'                 MOV     #FDBIN,R0               ;FDB ADDRESS
000122                                  CALL    .DLFNB                  ;CLOSE AND DELETE THE INPUT FOS FILE
000126  005267  001136                  INC     FOSCNT                  ;BUMP COUNT OF FOSSILS IN SUMMARY
000132  005767  001130                  TST     FOSBIN                  ;IS THIS THE FIRST INPUT FOS FILE?
000136  001014                         BNE     FOS.40                  ;BRANCH IF NOT
000140  012767  005652' 001120          MOV     #FOS1,FOSBIN            ;LOAD INCREMENTAL BUFFER ADDRESS
000146                                  FDBK$R  #FDBIN,FOSBIN           ;HAVE FCS SWITCH INPUT BUFFERS
000160  116767  001434  001470          MOVB    BCHNO,FOS0+4            ;INITIALIZE FOS HEADER WITH BATCH NO
000166  000413                         BR      FOS.50
                                ;
                                ;       ACCUMULATE THE FLU OCCURRENCES FROM INCREMENTAL BUFFER (FOS1)
                                ;       INTO THE SUMMARY BUFFER (FOS0)
                                ;
000170                          FOS.40:
000170  012703  001660'                 MOV     #FOS0+6,R3              ;INITIALIZE OUTPUT FOS POINTER
000174  012704  005662'                 MOV     #FOS1+8,R4              ;INITIALIZE INPUT FOS POINTER
000200  012302                          MOV     (R3)+,R2                ;GET QUERY COUNT
000202  003002                         BGT     10$                     ;BRANCH IF ANY QUERIES
000204                                  CALL    SUM.10                  ;REPORT BAD STRUCTURE AND ABORT THIS FOS
000210                          10$:
000210                                  CALL    FOSSUM                  ;ACCUMULATE FOS FOR A QUERY
000214  077203                         SOB     R2,10$                  ;NEXT QUERY
                                ;
000216                          FOS.50:
000216  026767  001036  001036          CMP     CSUN,LSUN               ;HAS LAST S.U. JUST BEEN PROCESSED?
000224  002046                         BGE     FOS.60                  ;IF SO, OUTPUT ACCUMULATED FOS
000226  005267  001026                  INC     CSUN                    ;UPDATE CURRENT SEARCH UNIT
000232  062767  000012  001024          ADD     #10,FDSCAD              ;UPDATE POINTER TO ITS FDSC
000240  000671                         BR      FOS.15                  ;PROCESS NEXT INPUT FILE
```

```
                                        ;
                                        ;       SUBROUTINE TO MERGE FOS COUNTERS FOR A SPECIFIC QUERY
                                        ;       FROM AN INPUT SEARCH UNIT INTO THE CUMULATIVE FOS COUNTERS
                                        ;       AT ENTRY (R3) -> QUERY ID FIELD (BEGINNING WORD) OF OUTPUT FOS
                                        ;                (R4) -> QUERY ID FIELD (BEGINNING WORD) OF INPUT FOS
                                        ;       AT EXIT, BOTH R3 AND R4 ARE UPDATED TO POINT TO THE NEXT ENTRY
                                        ;
000242                                  FOSSUM:
000242  022324                                  CMP     (R3)+,(R4)+             ;QUERY IDS CONSISTENT?
000244  001012                                  BNE     SUM.10                  ;BRANCH IF NOT
000246  021324                                  CMP     @R3,(R4)+               ;FLU COUNTS CONSISTENT?
000250  001010                                  BNE     SUM.10                  ;BRANCH IF NOT
000252  012305                                  MOV     (R3)+,R5                ;GET FLU COUNT FOR THIS ENTRY
000254  003405                                  BLE     20$                     ;BRANCH IF NO FLUS IN ENTRY
000256                                  10$:
000256  062423                                  ADD     (R4)+,(R3)+             ;SUM HIGH ORDER WORDS - STEP TO LOW ORDER WORDS
000260  062423                                  ADD     (R4)+,(R3)+             ;SUM LOW ORDER WORDS - STEP TO NEXT COUNTER PAIR
000262  005563  177774                          ADC     -4(R3)                  ;ADD IN POSSIBLE OVERFLOW
000266  077505                                  SOB     R5,10$                  ;NEXT FLU COUNTERS
000270                                  20$:
000270  000207                                  RETURN                          ;NORMAL EXIT POINT
                                        ;
                                        ;       INCOMPATIBLE FOS STRUCTURES - ABORT THIS SEARCH UNIT'S RESULTS
                                        ;
000272                                  SUM.10:
000272  016767  000762  000772                  MOV     CSUN.PAR1               ;SEARCH UNIT NUMBER
000300  005402                                  NEG     R2
000302  066702  001352                          ADD     FOS0+6,R2
000306  010267  000762                          MOV     R2,PAR2                 ;FOS ENTRY NUMBER
000312                                          MOUT$S  #MSG4,#PAR1             ;INCOMPATIBLE STRUCTURE MESSAGE
000332  005367  000732                          DEC     FOSCNT                  ;REDUCE COUNT OF FOSSILS DUE TO THIS ABOMINATION
000336  000167  177654                          JMP     FOS.50                  ;TRY NEXT INPUT FOS
```

```
                                        ;
                                        ;       ALL·FOS·FILES·FROM·THE SEARCH·UNITS·NOW·PROCESSED·
                                        ;       THE·MERGED·FOS·WILL·BE CHECKED·FOR·ANY·OVERFLOW·COUNTERS·
                                        ;       AND·WILL·THEN·BE·OUTPUT·TO·[7,4]FOS.MRG·
                                        ;
000342·                                 FOS.60:
000342·  005767  000722                         TST·    FOSCNT·                 ;ANY·FOS·REPORT·AT·ALL?·
000346·  001511                                 BEQ·    FOS.70                  ;IF·NOT, TELL·OPERATOR·SAD·TALE·AND·EXIT
000350·  012702·  001660'                       MOV·    #FOS0+6,R2·             ;R2·-> QUERY·COUNT·
000354·  012203                                 MOV·    (R2)+,R3                ;GET·QUERY COUNT·AND POINT·TO·FIRST·ENTRY·
000356                                  10$:
000356·  005722·                                TST·    (R2)+                   ;SKIP·OVER QUERY· ID·
000360·  012204                                 MOV·    (R2)+,R4                ;GET·FLU·COUNT·AND·POINT·TO·FIRST·COUNTER·
000362·                                 20$:
000362·  032722·  177774                       BIT·    #OFLMSK,(R2)+           ;DID·COUNTER·OVERFLOW?·
000366                                          BOFF·   30$                     ;BRANCH· IF NO·
000370·  012762·  000003  177776               MOV·    #HIFOS,-2(R2)           ;SET·HIGH·ORDER·WORD TO·MAX·VALUE·
000376·  012722·  177777                       MOV·    #-1,(R2)+               ;SET·LOW·ORDER·WORD·TO·MAX·VALUE·
000402·  000401                                 BR      40$
000404                                  30$:
000404·  005722·                                TST·    (R2)+                   ;POINT·TO·NEXT·COUNTER·
000406                                  40$:
000406·  077413                                 SOB·    R4,20$                  ;NEXT·COUNTER·
000410·  077316                                 SOB·    R3,10$                  ;NEXT·QUERY·
                                        ;
                                        ;       OUTPUT·MERGED·FOS·FILE
                                        ;
000412·  012700  001440'                        MOV·    #FDBOUT,R0              ;FDB·ADDRESS·
000416·  012701  000024                         MOV·    #FN.MFO,R1              ;FILE·NUMBER·
000422·                                         CALL·   BLDNFL·                 ;BUILD·FILE·NAME·BLOCK·IN·FDB·
000426                                          FDAT$R· #FDBOUT,,,,#-4          ;ALLOCATE·4 SECTORS·
                                        ;
000440                                          OFNB$W· #FDBOUT·
000456                                          WRITE$  #FDBOUT·
000466                                          WAIT$   #FDBOUT·
000476  103006                                  BCC·    50$
000500·  012767  177777  000552·                MOV·    #-1,CSUN·               ;MERGE·FILE·SIGNAL·
000506                                          CALL·   FCSERR·                 ;OUTPUT·ERROR·MESSAGE·
000512·  000427                                 BR      FOS.70                  ;TO·PROGRAM·EXIT·
000514                                  50$:
000514·  016767  001022  000000C·               MOV·    FDBOUT+F.FNB+N.FID,FDSC+FD.FID·
000522·  016767  001016  000000C·               MOV·    FDBOUT+F.FNB+N.FID+2,FDSC+FD.FID+2
000530·  016767  001024  000000C·               MOV·    FDBOUT+F.FNB+N.FVER,FDSC+FD.FVR·
000536·  012767  000024  000000C·               MOV·    #FN.MFO,FDSC+FD.FNB·
000544                                          CLOSE$  #FDBOUT·
000554                                          SDAT$C· HOTSK,RCVB·
000562·                                         RSUM$C· HOTSK·                  ;RESUME·HOTSK·
000570  000407                                  BR      FOS.80
```

```
    159 000572·                         FOS.70:
    160 000572·                                 MOUT$S·  #MSG3                  ;NO·MERGE·FILE·PRODUCED·
    161                                 ;
    162·                                ;       PROGRAM·EXIT POINT·
    163                                 ;
    164 000610                          FOS.80:
    165 000610                                  CALL·    GETFRE·                ;GET·PACKET·FOR·MSCHED·ACK·
    166 000614  016762· 001030  000002·         MOV·     ACKSCH,2(R2)           ;MOVE·IN·ACK·MESSAGE
    167 000622· 005062· 000004                  CLR·     4(R2)
    168 000626  116762· 000766  000004          MOVB·    BCHNO,4(R2)            ;MOVE·IN·BATCH·NO·
    169 000634                                  CALL·    PUTSSQ·                ;ENTER·PACKET·IN·THE·SSQ·
    170 000640                          FOS.90:
    171                                 ;       MOUT$S·  #MSG0
    172·000640                                  EXIT$S·
```

```
                                        .SBTTL  ERROR HANDLING ROUTINE
                                        .NLIST  BEX

                                ;
                                ;   DIRECTIVE ERROR
                                ;
000646  011667  000420          DIRERR: MOV     (SP),PAR1
000652  116700  0000A0G                 MOVB    $DSW,R0
000656  010067  000412                  MOV     R0,PAR2
000662                                  MOUT$S  #MSG1,#PAR1
000702  000207                          RETURN
                                ;
                                ;   FCS ERROR
                                ;
000704  011667  000362          FCSERR: MOV     (SP),PAR1
000710  016067  000052  000356          MOV     F.ERR(R0),PAR2
000716  112767  000377  000351          MOVB    #255,PAR2+1
000724  016767  000330  000344          MOV     CSUN,PAR3
000732                                  MOUT$S  #MSG2,#PAR1
000752  000207                          RETURN
                                ;
000754                                  .PSECT
                                ;
                                ;   STRING DESCRIPTORS
                                ;
000754  000024                  MSG0:   .WORD   LN0E-LN0
000756  001115'                         .WORD   LN0
000760  000040                  MSG1:   .WORD   LN1E-LN1
000762  001000'                         .WORD   LN1
000764  000055                  MSG2:   .WORD   LN2E-LN2
000766  001040'                         .WORD   LN2
000770  000025                  MSG3:   .WORD   LN3E-LN3
000772  001141'                         .WORD   LN3
000774  000071                  MSG4:   .WORD   LN4E-LN4
000776  001166'                         .WORD   LN4
                                ;
                                ;   FORMAT STRINGS
                                ;
001000      120     103     040 LN1:    .ASCIZ  /PC = %1O, DIRECTIVE ERROR = %1D/
001040                          LN1E:
001040      120     103     040 LN2:    .ASCIZ  /PC = %1O, FCS ERROR = %1D, SEARCH UNIT = %1D/
001115                          LN2E:
001115      106     117     123 LN0:    .ASCIZ  /FOSMRG PROGRAM EXIT/
001141                          LN0E:
001141      116     117     040 LN3:    .ASCIZ  /NO OUTPUT FOS REPORT/
001166                          LN3E:
001166      111     116     103 LN4:    .ASCIZ  /INCOMPATIBLE FOS STRUCTURES, S.U. = %1D, FOS ENTRY = %1D/
001257                          LN4E:
                                        .EVEN
```

```
                                        .SBTTL  DATA STORAGE FIELDS AND VARIABLES
                                ;
000003                          HIFOS   =       BIT0!BIT1
177774                          OFLMSK  =       ^C<HIFOS>
                                ;
001260  000000                  CSUN:   .WORD   0               ;CURRENT SEARCH UNIT NUMBER
001262  000001                  LSUN:   .WORD   N.SUNT-1        ;LAST SEARCH UNIT NUMBER
001264  00002G                  FDSCAD: .WORD   SUST+2          ;CURRENT FDSC POINTER
001266  000000                  FOSBIN: .WORD   0               ;INPUT BUFFER FLAG
001270  000000                  FOSCNT: .WORD   0               ;COUNT OF INPUT FOSSILS
                                ;
001272                          IOST:                           ;MSGOUT PARAMETER LIST
001272  000000                  PAR1:   .WORD   0
001274  000000                  PAR2:   .WORD   0
001276  000000                  PAR3:   .WORD   0
```

```
                                        .SBTTL  FILE STRUCTURES AND RECORD BUFFERS
                                ;
                                ;       FLU OCCURRENCE SUMMARY - SEARCH UNIT INPUT
                                ;
001300                          FDBIN:  FDBDF$
001440                                  FDRC$A  FD.RWM
001440                                  FDBK$A  FOS0,2*N.BUFW,,1,IOST
001440                                  FDOP$A  1
                                ;
                                ;       FLU OCCURRENCE SUMMARY - MERGED OUTPUT FILE
                                ;
001440                          FDBOUT: FDBDF$
001600                                  FDRC$A  FD.RWM
001600                                  FDBK$A  FOS0,2*N.BUFW,,1,IOST
001600                                  FDOP$A  1
001600                                  FSRSZ$  0
                                        .EVEN
                                ;
                                ;       HOTSK RCVQ BLOCK
                                ;
001600  007                     RCVB:   .BYTE   7                       ;PROGRAM ID
001601  000                             .BYTE   0                       ;FOS FILE INDICATOR
001602                          FDSC:   .BLKW   4                       ;FILE DESCRIPTOR AREA
                                ;
                                ;       MSCHED INPUT RCVB
                                ;
001612                          RCVIN:
001612                                  .BLKW   3
001620                          BCHNO:  .BLKW   1
001622                                  .BLKW   11
                                ;
                                ;       MSCHED ACK BLOCK
                                ;
001650  007                     ACKSCH: .BYTE   7                       ;PROGRAM ID
001651  001                             .BYTE   1                       ;TASK COMPLETE
                                ;
                                ;       FOS SUMMARY BUFFERS
                                ;
001652                          FOS0:   .BLKW   N.BUFW                  ;SUMMARY BUFFER
005652                          FOS1:   .BLKW   N.BUFW                  ;INCREMENTAL BUFFER
        000000'                         .END    FOSMRG
```

```
ACKSCH  001650R        B.NQRY  000232      010 FN.QLS  000006      011 F.RACC= 000016     PAR1    [illegible]01272R
BCHNO   001620R        B.QLSZ  000106      010 FN.QRY  000020      011 F.RATT= 000001     PAR2    001274R
BITVAL= 000000         B.QMAP  000234      010 FN.SF0  000030      011 F.RCNM= 000034     PAR3    001276R
BIT0   = 000001        B.QSPL  000316      010 FN.SF1  000032      011 F.RCTL= 000017     PUTSSQ= ****** GX
BIT1   = 000002        B.QTTM  000076      010 FN.SHD  000042      011 F.RSIZ= 000002     QE.RO1= 000144
BIT10  = 002000        B.QUQP  000056      010 FOSBIN  001266R         F.RTYP= 000000     Q.FDSC  000004     007
BIT11  = 004000        B.SFDB  000010      010 FOSCNT  001270R         F.SEQN= 000100     Q.NQBK  000000     007
BIT12  = 010000        B.SIZE  000772      010 FOSMRG  000000R         F.SPDV= 000072     Q.NUHL  000002     007
BIT13  = 020000        B.SNDP  000012      010 FOSSUM  000242R         F.SPUN= 000074     Q.SIZE  000014     007
BIT14  = 040000        B.SSQ   000004      010 FOS.10  000004R         F.STBK= 000036     RCVB    001600R
BIT15  = 100000        B.SSQF  000050      010 FOS.15  000024R         F.UNIT= 000136     RCVIN   001612R
BIT2   = 000004        B.STAT  000044      010 FOS.20  000066R         F.URBD= 000020     R.SUTN= 000002
BIT3   = 000010        B.STTE  000053      010 FOS.30  000116R         F.VBN = 000064     R.VXBA= 000006
BIT4   = 000020        B.UDOC  000110      010 FOS.40  000170R         F.VBSZ= 000060     R.VXTN= 000002
BIT5   = 000040        CF.B0  = 000070         FOS.50  000216R         GETFRE= ****** GX  SR.ARE  000114     002
BIT6   = 000100        CF.B2  = 000067         FOS.60  000342R         HIFOS = 000003     SR.ARS  000106     002
BIT7   = 000200        CF.B4  = 000066         FOS.70  000572R         IOST    001272R    SR.DAY  000010     002
BIT8   = 000400        CF.B6  = 000065         FOS.80  000610R         LN0     001115R    SR.DLT  000014     002
BIT9   = 001000        CF.DR0= 000064          FOS.90  000640R         LN0E    001141R    SR.ECB  000047     002
BLDEFL= ****** GX      CF.DR1= 000063          FOS0    001652R         LN1     001000R    SR.ECH  000046     002
BLDNFL= ****** GX      CSUN    001260R         FOS1    005652R         LN1E    001040R    SR.ECL  000050     002
BS.CLS= 000002         DBSLEN= 000116          FO.RD = ****** GX       LN2     001040R    SR.FIB  000012     002
BS.DBU= 000004         DH.BF0  000002      005 FO.WRT= ****** GX       LN2E    001115R    SR.GRE  000100     002
BS.INA= 000000         DH.BF1  000004      005 F.ACTL= 000076          LN3     001141R    SR.GRS  000072     002
BS.OPN= 000001         DH.CTL  000000      005 F.ALOC= 000040          LN3E    001166R    SR.LEN  000122     002
BS.SRC= 000003         DH.DMC  000010      005 F.BBFS= 000062          LN4     001166R    SR.LIN  000066     002
BYTE0  = 000000        DH.FLG  000006      005 F.BDB = 000070          LN4E    001257R    SR.LIP  000062     002
BYTE1  = 000001        DIRERR  000646R         F.BGBC= 000057          LSUN    001262R    SR.MON  000006     002
BYTE2  = 000002        DN.DCK  000000      013 F.BKDN= 000026          M      = 000062    SR.NDC  000042     002
BYTE3  = 000003        DN.NTP  000004      013 F.BKDS= 000020          MSGOUT= ****** GX  SR.NDS  000036     002
BYTE4  = 000004        DN.NXT  000006      013 F.BKEF= 000050          MSG0    000754R    SR.NIN  000030     002
BYTE5  = 000005        DN.ROT  000002      013 F.BKP1= 000051          MSG1    000760R    SR.NIP  000022     002
BYTE6  = 000006        DN.SIZ  000010      013 F.BKST= 000024          MSG2    000764R    SR.SDB  000032     002
BYTE7  = 000007        FCSERR  000704R         F.BKVB= 000064          MSG3    000770R    SR.SRC  000002     002
BYTE8  = 000010        FDBIN   001300R         F.CHR = 000075          MSG4    000774R    SR.SUN  000000     002
BYTE9  = 000011        FDBOUT  001440R         F.CNTG= 000034          N      = 000002    SR.TWS  000056     002
BYTVAL= 000012         FDSC    001602R         F.DFNB= 000046          N.BFAC= 000004     SR.WSL  000052     002
B.BSTA  000054     010 FDSCAD  001264R         F.DSPT= 000044          N.BHGH= 000006     SR.YR   000004     002
B.CNTX  000046     010 FD.FID  000000      003 F.DVNM= 000134          N.BTCH= 000004     SR.1IN  000024     002
B.CQUQ  000060     010 FD.FNB  000006      003 F.EFBK= 000010          N.BUFB= 004000     SR.1IP  000016     002
B.FEMA  000132     010 FD.FVR  000004      003 F.EFN = 000050          N.BUFW= 002000     SS.FID  000002     004
B.FEMB  000142     010 FD.LEN  000010      003 F.EOBB= 000032          N.DID = 000024     SS.FNB  000010     004
B.FEMC  000152     010 FD.RWM= ****** GX       F.ERR = 000052          N.DVNM= 000032     SS.FVR  000006     004
B.FFSA  000202     010 FN.DBR  000026      011 F.FACC= 000043          N.FID = 000000     SS.LEN  000012     004
B.FFSB  000212     010 FN.DBS  000022      011 F.FFBY= 000014          N.FNAM= 000006     SS.STT  000000     004
B.FFSC  000222     010 FN.DHR  000040      011 F.FNAM= 000110          N.FOS = 000764     ST.ASZ  000020     006
B.FMHR  000172     010 FN.EMA  000012      011 F.FNB = 000102          N.FTYP= 000014     ST.BSZ  000024     006
B.FQLS  000162     010 FN.EMB  000014      011 F.FTYP= 000116          N.FVER= 000016     ST.BTC  000000     006
B.FSAZ  000100     010 FN.EMC  000016      011 F.FVER= 000120          N.NEXT= 000022     ST.CSZ  000030     006
B.FSBZ  000102     010 FN.FSA  000000      011 F.HIBK= 000004          N.PKSZ= 000020     ST.HRL  000010     006
B.FSCZ  000104     010 FN.FSB  000002      011 F.LUN = 000042          N.PKTS= 000043     ST.LEN  000044     006
B.HBLK  000120     010 FN.FSC  000004      011 F.MBCT= 000054          N.QURY= 000031     ST.QRY  000002     006
B.HDOC  000114     010 FN.LGQ  000034      011 F.MBC1= 000055          N.STAT= 000020     ST.QSZ  000034     006
B.HRLP  000126     010 FN.LGU  000036      011 F.MBFG= 000056          N.SUNT= 000002     ST.SCH  000040     006
B.HRLR  000122     010 FN.MFO  000024      011 F.NRBD= 000024          N.UNI[illegible]4  ST.[illegible]VAL 000004 006
B.HRLW  000124     010 FN.MHR  000010      011 F.NREC= 000030          OFLMSK= 177774     ST.XLT  000014     006
B.NMBR  000052     010 FN.NMB  000044      011 F.OVBS= 000030          PAR$$$= 000027     SUM.10  000272R
```

```
FOSMRG PROGRAM  MACRO M1110  27-MAR-80 13:28  PAGE 17-2
SYMBOL TABLE
```

```
SUST.. = ****** GX      S.FNAM= 000006          WORD0 = 000000          XDBPRO= 000012          $$$OST= 000006
SU.DBU= 000004          S.FNB = 000036          WORD1 = 000002          XDMCIN= 000006          $$$T1 = 000005
SU.DON= 000006          S.FNBW= 000017          WORD2 = 000004          XFOSMR= 000007          .CLOSE= ****** G
SU.IDL= 000000          S.FNTY= 000004          WORD3 = 000006          XGTSRE= 000014          .DLFNB= ****** GX
SU.LOD= 000001          S.FTYP= 000002          WORD4 = 000010          XHITSK= 000011          .FINIT= ****** G
SU.SRC= 000002          S.HRL = 000240          WORD5 = 000012          XHLMER= 000002          .FSRCB= ****** G
SU.SRR= 000005          S.NFEN= 000020          WORD6 = 000014          XHOTSK= 000010          .OPFNB= ****** G
SU.XPD= 000003          WN.NTP  000004  012     WORD7 = 000016          XMSCHE= 000000          .READ = ****** G
S.BFHD= 000020          WN.NXT  000006  012     WORD8 = 000020          XOTS  = 000003          .WAIT = ****** G
S.DABA= 000006          WN.ROT  000002  012     WORD9 = 000022          XQT0  = 000001          .WRITE= ****** G
S.DAEF= 000010          WN.SIZ  000010  012     WRDVAL= 000024          XSULOA= 000005          ...PC1= 001440R
S.DATN= 000002          WN.SRC  000000  012     XBATCH= 000013          $DSW  = ****** GX       ...PC2= 001600R
S.FATT= 000016          WN.TYP  000001  012     XDBLOA= 000004          $$$   = 000022R   014   ...TPC= 000020
S.FDB = 000140
```

```
. ABS.  000000     000
        011652     001
SRCOFF  000122     002
FDSCOF  000010     003
SUSOFF  000012     004
DHROFF  000012     005
STTOFF  000044     006
QSPLOF  000014     007
BSTOFF  000772     010
FNOFFS  000044     011
WNODOF  000010     012
DNODOF  000010     013
$DPB$$  000030     014
$$FSR1  000000     015
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  6791 WORDS  (27 PAGES)
DYNAMIC MEMORY:  8084 WORDS  ( 31 PAGES)
ELAPSED TIME:  00:00:49
FOSMRG,FOSMRG/-SP=[20,1]P,M,FOSMRG
```

10- 2 REQUEST STATUS RECORD FROM SU
11- 40 GET PACKET, SEND DATA

```
                                        .TITLE  GTSREC
                                        .SBTTL  REQUEST STATUS RECORD FROM SU
                                ;
                                ;
                                        .MCALL  RCVX$C
                                        .MCALL  QIOW$S,MRKT$C,WTSE$C
                                ;
                                ;
          000006                COLUN=6
                                ;
                                ;
          000001                EF.IO=1
                                ;
                                ; LOCAL DATA
000000                          RDATA:  .BLKW   2       ;RECEIVE DATA AREA
000004  000000                          .WORD   0       ;COMMAND
000006  000000                          .WORD   0       ;SU#
000010                                  .BLKW   11.
                                ;
000036                          IOSTAT: .BLKW   2       ;I/O STATUS BLOCK
000042  000001                  XMLUN:  .WORD   1       ;LUN 1 =XM0
                                                        ;LUN 2 =XM1
                                ;
000044      122     123         RQSTSR: .ASCII  /RS/    ;REQUEST STATUS RECORD DMC EXCHANGE
000046                                  .BLKB   N.BUFB-2
                                ;
                                ;
                                ; ERROR MESSAGES
004044  000032                  ERR1:   .WORD   ERR1L
004046      107     124     123 ERR1T:  .ASCII  /GTSRECX: DMC WRITE FAILURE/
004051      122     105     103
004054      130     072     040
004057      104     115     103
004062      040     127     122
004065      111     124     105
004070      040     106     101
004073      111     114     125
004076      122     105
          000032                ERR1L=.-ERR1T
                                        .EVEN
                                ;
004100  000021                  ERR2:   .WORD   ERR2L
004102      107     124     123 ERR2T:  .ASCII  /GTSRECX: LINK INL/
004105      122     105     103
004110      130     072     040
004113      114     111     116
004116      113     040     111
004121      116     114
          000021                ERR2L=.-ERR2T
                                        .EVEN
                                ;
```

```
                                        .SBTTL  GET PACKET, SEND DATA
                                ;
004124                          GTSREC:
004124                                  RCVX$C  ,RDATA                  ;GET PACKET
                                ;
                                ;   SET UP LUN FOR UNIT INDICATED IN RECEIVE BUFFER
                                ;
004132                                  MRKT$C  EF,IO,30.,1             ;WAIT FOR 30 TICKS
004140                                  WTSE$C  EF,IO
004146  016767  173634  173666          MOV     RDATA+6,XMLUN
004154  062767  000001  173660          ADD     #1,XMLUN
                                ;
                                ; SEND THE EXCHANGE
                                ;
004162                          SNDDMC: QIOW$S  #IO,WLB,@#XMLUN,#EF,IO,,#IOSTAT,,<#RQSTSR,#N.BUFB>
004234  122767  000000G 173574          CMPB    #IS.SUC,IOSTAT          ;NEXT PACKET IF SUCCES
004242  001730                          BEQ     GTSREC
                                ; DETERMINE TYPE OF ERROR
004244  122767  000000G 173564          CMPB    #IE.CNR,IOSTAT
004252  001421                          BEQ     LNKRST
004254  122767  000000G 173554          CMPB    #IE.DNR,IOSTAT
004262  001415                          BEQ     LNKRST
004264  122767  000000G 173544          CMPB    #IE.TMO,IOSTAT
004272  001411                          BEQ     LNKRST
004274  122767  000000G 173534          CMPB    #IE.ABO,IOSTAT
004302  001405                          BEQ     LNKRST
                                ; UBRECOVERABLE LINK ERROR
004304  012700  004044'                 MOV     #ERR1,R0                ;MESSAGE TO CONSOLE
004310                                  CALL    COOUT
004314  000703                          BR      GTSREC                  ;NEXT PACKET
                                ;RECOVERABLE LINK ERROR
004316  012700  004100'          LNKRST: MOV     #ERR2,R0                ;MESSAGE TO CONSOLE
004322                                  CALL    COOUT
004326                                  MRKT$C  EF,IO,60.,1             ;WAIT FOR OTHER END TO RECOVER
004334                                  WTSE$C  EF,IO
004342  000707                          BR      SNDDMC                  ;RE-SEND EXCHANGE
                                ;
                                ; SEND MESSAGE TO CONSOLE
004344  012001                  COOUT:  MOV     (R0)+,R1
004346  116760  173434  000006          MOVB    RDATA+6,6(R0)
004354  152760  000060  000006          BISB    #60,6(R0)
004362                                  QIOW$S  #IO,WLB,#COLUN,#EF,IO,,,,<R0,R1,#40>
004430  000207                          RTS     PC
                                ;
                                ;
004124'                                 .END    GTSREC
```

SYMBOL TABLE

```
BITVAL= 000000          B.HDOC  000114      010 FD.LEN  000010      003 Q.NQBK  000000      007 ST.SCH  000040      006
BIT0  = 000001          B.HRLP  000126      010 FN.DBR  000026      011 Q.NUHL  000002      007 ST.UHL  000004      006
BIT1  = 000002          B.HRLR  000122      010 FN.DBS  000022      011 Q.SIZE  000014      007 ST.XLT  000014      006
BIT10 = 002000          B.HRLW  000124      010 FN.DHR  000040      011 RDATA   000000R         SU.DBU= 000004
BIT11 = 004000          B.NMBR  000052      010 FN.EMA  000012      011 RQSTSR  000044R         SU.DON= 000006
BIT12 = 010000          B.NQRY  000232      010 FN.EMB  000014      011 R.VXBA= 000006          SU.IDL= 000000
BIT13 = 020000          B.QLSZ  000106      010 FN.EMC  000016      011 R.VXTN= 000002          SU.LOD= 000001
BIT14 = 040000          B.QMAP  000234      010 FN.FSA  000000      011 SNDDMC  004162R         SU.SRC= 000002
BIT15 = 100000          B.QSPL  000316      010 FN.FSB  000002      011 SR.ARE  000114      002 SU.SRR= 000005
BIT2  = 000004          B.QTTM  000076      010 FN.FSC  000004      011 SR.ARS  000106      002 SU.XPD= 000003
BIT3  = 000010          B.QUQP  000056      010 FN.LGD  000034      011 SR.DAY  000010      002 S.HRL = 000240
BIT4  = 000020          B.SFDB  000010      010 FN.LGU  000036      011 SR.DLT  000014      002 WN.NTP  000004      012
BIT5  = 000040          B.SIZE  000772      010 FN.MFO  000024      011 SR.ECB  000047      002 WN.NXT  000006      012
BIT6  = 000100          B.SNDP  000012      010 FN.MHR  000010      011 SR.ECH  000046      002 WN.ROT  000002      012
BIT7  = 000200          B.SSQ   000004      010 FN.NMB  000044      011 SR.ECL  000050      002 WN.SIZ  000010      012
BIT8  = 000400          B.SSQF  000050      010 FN.QLS  000006      011 SR.FIB  000012      002 WN.SRC  000000      012
BIT9  = 001000          B.STAT  000044      010 FN.QRY  000020      011 SR.FIB  000012      002 WN.TYP  000001      012
BS.CLS= 000002          B.STTE  000053      010 FN.SF0  000030      011 SR.GRE  000100      002 WORD0 = 000000
BS.DBU= 000004          B.UDOC  000110      010 FN.SF1  000032      011 SR.GRS  000072      002 WORD1 = 000002
BS.INA= 000000          CF.B0 = 000070          FN.SHD  000042      011 SR.LEN  000122      002 WORD2 = 000004
BS.OPN= 000001          CF.B2 = 000067          GTSREC  004124R     011 SR.LIN  000066      002 WORD3 = 000006
BS.SRC= 000003          CF.B4 = 000066          IE.ABO= ****** GX       SR.LIP  000062      002 WORD4 = 000010
BYTE0 = 000000          CF.B6 = 000065          IE.CNR= ****** GX       SR.MON  000006      002 WORD5 = 000012
BYTE1 = 000001          CF.DR0= 000064          IE.DNR= ****** GX       SR.NDC  000042      002 WORD6 = 000014
BYTE2 = 000002          CF.DR1= 000063          IE.TMO= ****** GX       SR.NDS  000036      002 WORD7 = 000016
BYTE3 = 000003          COLUN = 000006          IOSTAT  000036R         SR.NIN  000030      002 WORD8 = 000020
BYTE4 = 000004          COOUT   004344R         IO.WLB= ****** GX       SR.NIP  000022      002 WORD9 = 000022
BYTE5 = 000005          DBSLEN= 000116          IS.SUC= ****** GX       SR.SDB  000032      002 WRDVAL= 000024
BYTE6 = 000006          DH.BF0  000002      005 LNKRST  004316R         SR.SRC  000002      002 W.TSEF= 000002
BYTE7 = 000007          DH.BF1  000004      005 M       = 000062        SR.SUN  000000      002 XBATCH= 000013
BYTE8 = 000010          DH.CTL  000000      005 M.KTAE= 000010          SR.TWS  000056      002 XDBLOA= 000004
BYTE9 = 000011          DH.DMC  000010      005 M.KTEF= 000002          SR.WSL  000052      002 XDBPRO= 000012
BYTVAL= 000012          DH.FLG  000006      005 M.KTMG= 000004          SR.YR   000004      002 XDMCIN= 000006
B.BSTA  000054      010 DN.DCK  000000      013 M.KTUN= 000006          SR.1IN  000024      002 XFOSMR= 000007
B.CNTX  000046      010 DN.NTP  000004      013 N       = 000002        SR.1IP  000016      004 XGTSRE= 000014
B.COUQ  000060      010 DN.NXT  000006      013 N.BFAC= 000004          SS.FID  000002      004 XHITSK= 000011
B.FEMA  000132      010 DN.ROT  000002      013 N.BHGH= 000006          SS.FNB  000010      004 XHLMER= 000002
B.FEMB  000142      010 DN.SIZ  000010      013 N.BTCH= 000004          SS.FVR  000006      004 XHOTSK= 000010
B.FEMC  000152      010 EF.IO = 000001          N.BUFB= 004000          SS.LEN  000012      004 XMLWN   000042R
B.FFSA  000202      010 ERRJ    004044R         N.BUFW= 002000          SS.STT  000000      006 XMSCHE= 000000
B.FFSB  000212      010 ERR1L = 000032          N.FOS = 000764          ST.ASZ  000020      006 XQTS  = 000003
B.FFSC  000222      010 ERR1T   004046R         N.PKSZ= 000020          ST.BSZ  000024      006 XQT0  = 000001
B.FMHR  000172      010 ERR2    004100R         N.PKTS= 000043          ST.BTC  000000      006 XSULOA= 000005
B.FQLS  000162      010 ERR2L = 000021          N.QURY= 000031          ST.CSZ  000030      006 $$$   = 000040R     014
B.FSAZ  000100      010 ERR2T   004102R         N.SUNT= 000002          ST.HRL  000010      006 $$$ADC= 000002
B.FSBZ  000102      010 FD.FID  000000      003 QE.RQ1= 000144          ST.LEN  000044      006 $$$OST= 000004
B.FSCZ  000104      010 FD.FNB  000006      003 Q.FDSC  000004          ST.QRY  000002      006 $$$T1 = 000000
B.HBLK  000120      010 FD.FVR  000004      003                         ST.QSZ  000034
```

```
. ABS.  000000     000
        004432     001
SRCOFF  000122     002
FDSCOF  000010     003
SUSOFF  000012     004
DHROFF  000012     005
STTOFF  000044     006
QSPLOF  000014     007
```

```
GTSREC· M[redacted]0·M1110  27-MAR-80 13:25  PAGE·11-2·
SYMBOL·TA[redacted].

BSTOFF··000772·     010
FNOFFS·  000044     011
UNODOF·  000010     012·
DNODOF·  000010     013
$DPB$$   000044     014
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  3860 WORDS·  (·16 PAGES)
DYNAMIC·MEMORY:  4916 WORDS·  ( 18 PAGES)
ELAPSED·TIME:  00:00:26
GTSREC›GTSREC/-SP=[20,1]P›M›GTSREC·
```

STATUS· M[redacted]0·M1110 27-MAR-80 13:37
TABLE·OF·C[redacted]NTENTS·

10- 2· STT·COMMAND·DISPATCHER·

```
                                        .TITLE··STATUS·
                                        .SBTTL· STT·COMMAND·DISPATCHER·
                                ;
                                ;
                                        .MCALL· GMCR$,DIR$,EXIT$S·
                                        .MCALL· SDAT$S,ROST$S,QIOW$S·
                                ;
                                ;
                                        .GLOBL· $DSW
                                ;
                                ;
000005                          TILUN=5
000001                          TIEF=1
                                ;
```

```
                                        ;
                                        ; DATA·
000000                                  GETCMD:·GMCR$                   ;GET·COMMAND·LINE·
                                        ;
                                        ;
                                        ; MESSAGES
000122· 000024                          MSG1:   .WORD·  MSG1L·
000124      123     124     124         MSG1T:  .ASCII· /STT: INVALID·COMMAND/·
000127      072·    040     111
000132·     116     126     101
000135      114     111     104
000140      040     103     117
000143      115     115     101
000146      116     104
          000024                        MSG1L=.-MSG1T·
                                                .EVEN·
                                        ;
                                        ;
000150                                  SNDPKT: .BLKW·  13.             ;SEND·PACKET·
000202·                                 IOSTAT: .BLKW·  2·              ;I/O·STATUS·
                                        ;
                                        ;
                                        ; COMMAND·DISPATCH·TABLE·
000206                                  DSPTBL:
000206  000240'                                  .WORD·  BTCHST·         ;ADDRESS OF·STRING
000210  000000                                  .WORD·  0               ;COMMAND CODE·
000212· 006274  012000                          .RAD50  /BATCH/·        ;HANDLING·TASK·NAME·
000216  000246'                                  .WORD·  DTBSST·
000220  000002·                                 .WORD·  2·
000222· 014543  076474                          .RAD50  /DBSTAT/·
000226  000257'                                 .WORD·  STTSST·
000230  000004                                  .WORD·  4
000232· 074741  077770                          .RAD50  /STATS/·
                                        ;
000236  000000                                  .WORD·  0               ;END·OF·TABLE·
                                        ;
                                        ; COMMAND·CODE·STRINGS·
000240      102·    101     124     BTCHST: .ASCIZ· /BATCH/·
000243      103     110     000
000246      104     101     124     DTBSST: .ASCIZ· /DATABASE/·
000251      101     102     101
000254      123     105     000
000257      123     124     101     STTSST: .ASCIZ· /STATS/·
000262·     124     123     000
                                                .EVEN·
                                        ;
```

```
                                        ;
                                        ; DETERMINE·TYPE·OF·INPUT·
                                        ;
000266                                  START:
000266                                          DIR$    #GETCMD·                ;GET·COMMAND·LINE·
000274  016701  000000G·                        MOV·    $DSW,R1                 ;LENGTH·OF·DATA·
000300  162701  000004                          SUB·    #4,R1                   ;MINUS·'STT·'
000304  003437                                  BLE·    SYNTXE·                 ;NO·PARAMS
                                        ; START·COMMAND·PARSE·
000306                                  PARSIN: SAVE·   R1                      ;LENGTH·OF·PARAMS·
000310  012704  000206'                        MOV·    #DSPTBL,R4              ;FIRST·DISPATCH·TABLE·ENTRY·
000314                                          SAVE·   R4
                                        ; NEXT·DISPATCH·TABLE·ENTRY·
000316  012702· 000006'                  1$:     MOV·    #GETCMD+G.MCRB+4,R2·    ;ADDRESS·OF·COMMAND·
000322· 016601  000002                          MOV·    2(SP),R1                ;LENGTH·OF·INPUT·
000326  011604                                  MOV·    (SP),R4                 ;NEXT·DISP·TABLE·ENTRY·
000330  012403                                  MOV·    (R4)+,R3                ;COMMAND·STRING·ADDRESS·
                                        ;
000332· 122322·                         2$:     CMPB·   (R3)+,(R2)+             ;BRANCH·IF·COMMAND·
000334  001014                                  BNE·    4$                      ;  STRING·MISMATCH·
000336  122712· 000057                          CMPB·   #'/,(R2)                ;MATCH·-·INPUT·DELIMITER?·
000342· 001006                                  BNE·    3$                      ;NO·-·CONTINUE·
000344  105713                                  TSTB·   (R3)                    ;COMMAND·IN·TABLE·DELIMITED·
000346  001004                                  BNE·    3$                      ;NO·-·CONTINUE·
000350  162701  000002                          SUB·    #2,R1                   ;MATCHED·-·ACCOUNT·FOR·'/'
000354  005202·                                 INC·    R2·
000356  000443                                  BR      MTCHCD·
000360  077114                          3$:     SOB·    R1,2$                   ;CONTINUE·COMMAND·SCAN·
000362· 105713                                  TSTB·   (R3)                    ;OUT·OF·COMMAND·INPUT·-·BRANCH
000364· 001440                                  BEQ·    MTCHCD·                 ;  IF·END·OF·COMMAND
                                        ; NO·MATCH·THIS·ENTRY·
000366  062716  000010                  4$:     ADD·    #8.,(SP)                ;NEXT·TABLE·ENTRY·ADDRESS·
000372· 005776  000000                          TST·    @0(SP)                  ;CONTINUE·IF·NOT·
000376  001347                                  BNE·    1$                      ;  END·OF·TABLE·
                                        ; NO·MATCH·IN·TABLE·
000400  062706  000004                          ADD·    #4,SP·                  ;RESTORE·STACK·
                                        ;
                                        ; SYNTAX·ERROR·
000404  012700  000122'                 SYNTXE: MOV·    #MSG1,R0
                                        ;
                                        ; ISSUE·MESSAGE·TO·CONSOLE·
000410  012001                          COOUT:  MOV·    (R0)+,R1
000412·                                         QIOW$S· #IO.WLB,#TILUN,#TIEF,,,,<R0,R1,#40>
                                        ;
000460                                          EXIT$S·
                                        ;
                                        ;
                                        ; MATCHED·DISPATCH·TABLE·ENTRY
000466  062706  000004                  MTCHCD: ADD·    #4,SP·                  ;RESTORE·STACK·
000472· 012700  000150'                         MOV·    #SNDPKT,R0              ;SEND·PACKET·ADDRESS
000476  012420                                  MOV·    (R4)+,(R0)+             ;COMMAND·CODE·
000500  010120                                  MOV·    R1,(R0)+                ;#·OF·PARAM·CHARS·
000502· 001406                                  BEQ·    2$                      ;NO·PARAMS
000504  100737                                  BMI·    SYNTXE·                 ;ERROR·SLIPPED·THROUGH·
000506  020127  000026                          CMP·    R1,#22.                 ;ERROR·IN·22·
000512· 101334                                  BHI·    SYNTXE·
000514  112220                          1$:     MOVB·   (R2)+,(R0)+             ;PARAMS·TO·PACKET·
```

```
    110 000516  077102·                     SOB·     R1,1$
    111                                 ;
    112·000520                          2$:  SDAT$S·  R4,#SNDPKT·          ;SEND DATA
    113 000542·                              RQST$S·  R4                   ;RUN TASK·
    114                                 ;
    115 000566                               EXIT$S·                       ;DONE·
    116                                 ;
    117                                 ;
    118                                 ;
    119         000266'                      .END·    START·
```

```
BITVAL= 000000          B.FSBZ  000102          010 FN.DBR  000026          011 SNDPKT  000150R         SU.DBU= 000004
BIT0  = 000001          B.FSCZ  000104          010 FN.DBS  000022          011 SR.ARE  000114      002 SU.DON= 000006
BIT1  = 000002          B.HBLK  000120          010 FN.DHR  000040          011 SR.ARS  000106      002 SU.IDL= 000000
BIT10 = 002000          B.HDOC  000114          010 FN.EMA  000012          011 SR.DAY  000010      002 SU.LOD= 000001
BIT11 = 004000          B.HRLP  000126          010 FN.EMB  000014          011 SR.DLT  000014      002 SU.SRC= 000002
BIT12 = 010000          B.HRLR  000122          010 FN.EMC  000016          011 SR.ECB  000047      002 SU.SRR= 000005
BIT13 = 020000          B.HRLW  000124          010 FN.FSA  000000          011 SR.ECH  000046      002 SU.XPD= 000003
BIT14 = 040000          B.NMBR  000052          010 FN.FSB  000002          011 SR.ECL  000050      002 SYNTXE  000404R
BIT15 = 100000          B.NQRY  000232          010 FN.FSC  000004          011 SR.FIB  000012      002 S.HRL = 000240
BIT2  = 000004          B.QLSZ  000106          010 FN.LGO  000034          011 SR.GRE  000100      002 TIEF  = 000001
BIT3  = 000010          B.QMAP  000234          010 FN.LGU  000036          011 SR.GRS  000072      002 TILUN = 000005
BIT4  = 000020          B.QSPL  000316          010 FN.MFO  000024          011 SR.LEN  000122      002 WN.NTP  000004      012
BIT5  = 000040          B.QTTM  000076          010 FN.MHR  000010          011 SR.LIN  000066      002 WN.NXT  000006      012
BIT6  = 000100          B.QUQP  000056          010 FN.NMB  000044          011 SR.LIP  000062      002 WN.ROT  000002      012
BIT7  = 000200          B.SFDB  000010          010 FN.QLS  000006          011 SR.MON  000006      002 WN.SIZ  000010      012
BIT8  = 000400          B.SIZE  000772          010 FN.QRY  000020          011 SR.NDC  000042      002 WN.SRC  000000      012
BIT9  = 001000          B.SNDP  000012          010 FN.SF0  000030          011 SR.NDS  000036      002 WN.TYP  000001      012
BS.CLS= 000002          B.SSQ   000004          010 FN.SF1  000032          011 SR.NIN  000030      002 WORD0 = 000000
BS.DBU= 000004          B.SSQF  000050          010 FN.SHD  000042          011 SR.NIP  000022      002 WORD1 = 000002
BS.INA= 000000          B.STAT  000044          010 GETCMD  000000R             SR.SDB  000032      002 WORD2 = 000004
BS.OPN= 000001          B.STTE  000053          010 G.MCRB= 000002              SR.SRC  000002      002 WORD3 = 000006
BS.SRC= 000003          B.UDOC  000110          010 IOSTAT  000202R             SR.SUN  000000      002 WORD4 = 000010
BTCHST  000240R         CF.B0 = 000070              IO.WLB= ****** GX           SR.TWS  000056      002 WORD5 = 000012
BYTE0 = 000000          CF.B2 = 000067              M     = 000062              SR.WSL  000052      002 WORD6 = 000014
BYTE1 = 000001          CF.B4 = 000066              MSG1      000122R           SR.YR   000004      002 WORD7 = 000016
BYTE2 = 000002          CF.B6 = 000065              MSG1L = 000024              SR.1IN  000024      002 WORD8 = 000020
BYTE3 = 000003          CF.DR0= 000064              MSG1T   000124R           SR.1IP  000016      002 WORD9 = 000022
BYTE4 = 000004          CF.DR1= 000063              MTCHCD  000466R           SS.FID  000002      004 WRDVAL= 000024
BYTE5 = 000005          COOUT   000410R             N     = 000002            SS.FNB  000010      004 XBATCH= 000013
BYTE6 = 000006          DBSLEN= 000116              N.BFAC= 000004            SS.FVR  000006      004 XDBLOA= 000004
BYTE7 = 000007          DH.BF0  000002          005 N.BHGH= 000006            SS.LEN  000012      004 XDBPRO= 000012
BYTE8 = 000010          DH.BF1  000004          005 N.BTCH= 000004            SS.STT  000000      004 XDMCIN= 000006
BYTE9 = 000011          DH.CTL  000000          005 N.BUFB= 004000            START   000266R             XFOSMR= 000007
BYTVAL= 000012          DH.DMC  000010          005 N.BUFW= 002000            STTSST  000257R             XGTSRE= 000014
B.BSTA  000054      010 DH.FLG  000006          005 N.FOS = 000764            ST.ASZ  000020      006 XHITSK= 000011
B.CNTX  000046      010 DH.DCK  000000          013 N.PKSZ= 000020            ST.BSZ  000024      006 XHLMER= 000002
B.CQUQ  000060      010 DN.NTP  000004          013 N.PKTS= 000043            ST.BTC  000000      006 XHOTSK= 000010
B.FEMA  000132      010 DN.NXT  000006          013 N.QURY= 000031            ST.CSZ  000030      006 XMSCHE= 000000
B.FEMB  000142      010 DN.ROT  000002          013 N.SUNT= 000002            ST.HRL  000010      006 XQTS  = 000003
B.FEMC  000152      010 DN.SIZ  000010          013 PARSIN  000306R           ST.LEN  000044      006 XQT0  = 000001
B.FFSA  000202      010 DSPTBL  000206R             QE.ROI= 000144            ST.QRY  000002      006 XSULOA= 000005
B.FFSB  000212      010 DTBSST  000246R             Q.FDSC  000004      007 ST.QSZ  000034      006 $DSW  = ****** G
B.FFSC  000222      010 FD.FID  000000          003 Q.NQBK  000000      007 ST.SCH  000040      006 $$$ARG= 000002
B.FMHR  000172      010 FD.FNB  000006          003 Q.NUHL  000002      007 ST.UHL  000004      006 $$$OST= 000122
B.FQLS  000162      010 FD.FVR  000004          003 Q.SIZE  000014      007 ST.XLT  000014      006 $$$T2 = 000004
B.FSAZ  000100      010 FD.LEN  000010          003

. ABS.  000000      000
        000574      001
SRCOFF  000122      002
FDSCOF  000010      003
SUSOFF  000012      004
DHRCFF  000012      005
STTOFF  000044      006
QSPLOF  000014      007
BSTOFF  000772      010
FNOFFS  000044      011
```

```
STATUS  M[illegible]O M1110  27-MAR-80 13:37  PAGE 12-3
SYMBOL TA[illegible]E

WNODOF  000010     012
DNODOF  000010     013
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3504 WORDS  ( 14 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:23
STATUS,STATUS/-SP=[20,1]P,M,STATUS
```

BATCH· M[redacted]D·M1110 27-MAR-80 13:37
TABLE·OF·C[redacted]TENTS·

10- 2· DISPLAY·BATCH·STATUS·

```
                                        .TITLE  BATCH
                                        .SBTTL  DISPLAY BATCH STATUS
                                ;
                                ;
                                        .MCALL  RCVX$C,EXIT$S,QIOW$S
                                ;
                                ;
                                        .GLOBL  BSTPTR
                                ;
                                ;
000005                          TILUN=5
000001                          TIEF=1
                                ;
```

```
                                        ;
                                        ; DATA·
                                        ;
000000                                  RCVDAT: .BLKW·  2·      ;RECEIVE·PACKET·
000004                                  RCVCMD: .BLKW·  1       ;COMMAND·CODE·
000006                                  RCVCHR: .BLKW·  1       ;#·OF·PARAMETER·CHARACTERS
000010                                  RCVPRM: .BLKB·  22.     ;PARAMETER·TEXT·
                                        ;
000036                                  IOSTAT: .BLKW·  2·
000042·                                 SAVESP: .BLKW·  1
                                        ;
                                        ;
                                        ; TABLE·HEADING·
                                        ;
000044  000043                          MSGi:   .WORD·  MSGiL·
000046      102·    101     124         MSGiT:  .ASCII· /BATCH· STATE·              QUERIES/·
000051      103     110     040
000054      040     123     124
000057      101     124     105
000062·     040     040     040
000065      040     040     040
000070      040     040     040
000073      040     040     040
000076      040     040     040
000101      040     121     125
000104      105     122     111
000107      105     123
          000043                        MSGiL=.-MSGiT·
                                        .EVEN·
                                        ;
                                        ;
                                        ; BATCH·STATUS·LINE·SKELETON·
                                        ;
000112·     040                         SKELTN: .ASCII· /·/·
000113                                  BNO:    .BLKB·  2·
000115      040     040     040                 .ASCII· /·   /·
000120      040
000121                                  STATE:  .BLKB·  20.
          000033                        SKLSHT=.-SKELTN·
000145·     040     040                 .ASCII· /· /
000147                                  NQR:    .BLKB·  3
          000040                        SKLNGT=.-SKELTN·
                                        .EVEN·
                                        ;
                                        ;
                                        ;
                                        ; BATCH·STATE·MESSAGE·TABLES·
                                        ;
                                        ; INDEX·BATCH·STATE·TO·MESSAGE INDEX·
                                        ;  BYTE·0 - INDEX·
                                        ;  BYTE·1 - BIT0: HAS·BATCH·STATE·STATUS INFORMATION·
                                        ;            BIT1: NEEDS·#·OF·QUERIES·
                                        ;
000152·     000     000         BSIXIX: .BYTE·  0,0             ;BS.INA·
000154      002·    003                 .BYTE·  2,BIT0!BIT1     ;BS.OPN·
000156      006     002                 .BYTE·  6,BIT1          ;BS.CLS·
000160      010     002                 .BYTE·  8.,BIT1         ;BS.SRC·
```

```
 60 000162     012     001                    .BYTE   10.,BIT0          ;BS.DBU
 61                                           ;
 62                                           ; STATE MESSAGE INDEX TABLE
 63                                           ;
 64 000164  000202'                           BSTINX: .WORD   BS1
 65 000166  000213'                                   .WORD   BS2
 66 000170  000227'                                   .WORD   BS3
 67 000172  000246'                                   .WORD   BS4
 68 000174  000255'                                   .WORD   BS5
 69 000176  000264'                                   .WORD   BS6
 70 000200  000311'                                   .WORD   BS7
 71                                           ;
 72                                           ; STATE MESSAGES
 73                                           ;
 74 000202     111     116     101  BS1:     .ASCIZ  /INACTIVE/
    000205     103     124     111
    000210     126     105     000
 75 000213     117     120     105  BS2:     .ASCIZ  *OPEN/ACTIVE*
    000216     116     057     101
    000221     103     124     111
    000224     126     105     000
 76 000227     117     120     105  BS3:     .ASCIZ  *OPEN/TRANSLATE*
    000232     116     057     124
    000235     122     101     116
    000240     123     114     101
    000243     124     105     000
 77 000246     103     114     117  BS4:     .ASCIZ  /CLOSED/
    000251     123     105     104
    000254     000
 78 000255     123     105     101  BS5:     .ASCIZ  /SEARCH/
    000260     122     103     110
    000263     000
 79 000264     104     101     124  BS6:     .ASCIZ  *DATA BASE/CONTINUOUS*
    000267     101     040     102
    000272     101     123     105
    000275     057     103     117
    000300     116     124     111
    000303     116     125     117
    000306     125     123     000
 80 000311     104     101     124  BS7:     .ASCIZ  *DATA BASE/MASS*
    000314     101     040     102
    000317     101     123     105
    000322     057     115     101
    000325     123     123     000
 81                                           ;
 82 000330     040     040     040  BLANKS:  .ASCII  /                    /  ;BLANK FILL
    000333     040     040     040
    000336     040     040     040
    000341     040     040     040
    000344     040     040     040
    000347     040     040     040
    000352     040     040
 83                                           .EVEN
 84                                           ;
```

```
                                        ;
                                        START:
000354  010667  177462                          MOV     SP,SAVESP               ;SAVE IDLE TASK STACK POINTER
                                        ;
000360  016706  177456                  RSTART: MOV     SAVESP,SP               ;RESTORE STACK
000364                                          RCVX$C  ,RCVDAT                 ;TRY FOR PACKET
000372  103003                                  BCC     1$
000374                                          EXIT$S
                                        ;
000402  012700  000044'                 1$:     MOV     #MSG1,R0                ;PUT OUT TABLE HEADER
000406  012001                                  MOV     (R0)+,R1
000410                                          CALL    TIOUT
                                        ;
                                        ; OUTPUT STATUS LINE FOR EACH BATCH
000414  005004                                  CLR     R4                      ;BATCH 0
                                        ; NEXT BATCH
000416  016405  000000G                 NXTBTC: MOV     BSTPTR(R4),R5           ;BST ADDRESS
000422  010401                                  MOV     R4,R1                   ;MOVE ASCII BATCH NUMBER
000424  005000                                  CLR     R0                      ;  TO MESSAGE WITH LEADING ZERO
000426  071027  000012                          DIV     #10.,R0
000432  062701  000060                          ADD     #'0,R1
000436  110167  177452                          MOVB    R1,BNO+1
000442  062700  000060                          ADD     #'0,R0
000446  110067  177441                          MOVB    R0,BNO
                                        ; BUILD BATCH STATE MESSAGE
000452  116500  000053                          MOVB    B.STTE(R5),R0           ;BATCH STATE
000456  060000                                  ADD     R0,R0                   ;INDEX
000460  116001  000152'                         MOVB    BSIXIX(R0),R1           ;MESSAGE INDEX
000464  132760  000001  000153'                 BITB    #BIT0,BSIXIX+1(R0)      ;BRANCH IF NO SUB STATE FLAGS
000472  001406                                  BEQ     1$
000474  032765  000001  000054                  BIT     #BIT0,B.BSTA(R5)        ;PLUS 2 IF SUBSTATE SHOWS OTHER STATUS
000502  001402                                  BEQ     1$
000504  062701  000002                          ADD     #2,R1
000510  016101  000164'                 1$:     MOV     BSTINX(R1),R1           ;ADDRESS OF TEXT
000514  012702  000121'                         MOV     #STATE,R2               ;PLACE IN MESSAGE
000520  012703  000024                          MOV     #20.,R3                 ;FIELD WIDTH
000524  112122                        STATXT: MOVB    (R1)+,(R2)+             ;TEXT CHAR
000526  001005                                  BNE     1$                      ;NOT END OF TEXT
000530  112762  000040  177777                  MOVB    #' ,-1(R2)              ;RECOVER BLANK
000536  012701  000330'                         MOV     #BLANKS,R1              ;BLANK FILL REST
000542  077310                          1$:     SOB     R3,STATXT
                                        ; PUT IN # OF QUERIES
000544  012701  000033                  NMBQTX: MOV     #SKLSHT,R1              ;ASSUME NO QUERY #
000550  132760  000002  000153'                 BITB    #BIT1,BSIXIX+1(R0)      ;BRANCH IF SO
000556  001424                                  BEQ     2$
000560  112767  000040  177361                  MOVB    #' ,NQR                 ;INIT HIGH ORDER BLANKS
000566  112767  000040  177354                  MOVB    #' ,NQR+1
000574  012702  000152'                         MOV     #NQR+3,R2               ;START A LOW ORDER
000600  016501  000232                          MOV     B.NQRY(R5),R1           ;# OF QUERIES
000604  005000                          1$:     CLR     R0                      ;SPLIT OFF LOW ORDER DECADE
000606  071027  000012                          DIV     #10.,R0
000612  062701  000060                          ADD     #'0,R1                  ;MAKE IT ASCII
000616  110142                                  MOVB    R1,-(R2)                ;INTO MESSAGE
000620  010001                                  MOV     R0,R1                   ;CONTINUE TILL ZERO
000622  001370                                  BNE     1$
000624  012701  000040                          MOV     #SKLNGT,R1              ;FULL LENGTH OF TEXT
000630  012700  000112'                 2$:     MOV     #SKELTN,R0              ;TEXT START
```

```
BATCH  MACRO M1110  27-MAR-80 13:37  PAGE 12-1
DISPLAY BATCH STATUS

     143 000634                                 CALL    TIOUT                   ;OUTPUT
     144                                ;
     145 000640  062704  000002                 ADD     #2,R4                   ;DO ALL BATCHES
     146 000644  020427  000006                 CMP     R4,#N.BHGH
     147 000650  101662                         BLOS    NXTBTC
     148 000652  000167  177502                 JMP     RSTART                  ;NEXT PACKET
     149                                ;
```

```
                                        ;
                                        ; OUTPUT·A BATCH·STATUS·LINE·
                                        ;
                                        ; INPUT·- R0: DATA·START·ADDRESS·
                                        ;         R1: LENGTH OF·TEXT·
                                        ;
000656                                  TIOUT:  QIOW$S·  #IO,#LB,#TILUN,#TIEF,,,#IOSTAT,,<R0,R1,#40>
000726  122767  000000G·177102·               CMPB·   #IS.SUC,IOSTAT
000734  001001                          BNE·    PRCERR·
000736                                  RTN·
                                        ;
                                        ; ERROR·
                                        ;
000740                                  PRCERR:
000740  000167  177414                  JMP·    RSTART·
                                        ;
                                        ;
        000354'                         .END·   START·
```

```
BITVAL= 000000          B.FEMC  000152          010 FN.DBR  000026      011 RCVPRM  000010R         ST.OSZ  000034      006
BIT0  = 000001          B.FFSA  000202          010 FN.DBS  000022      011 RSTART  000360R         ST.SCH  000040      006
BIT1  = 000002          B.FFSB  000212          010 FN.DHR  000040      011 R.VXBA= 000006          $T.UHL  000004      006
BIT10 = 002000          B.FFSC  000222          010 FN.EMA  000012      011 R.VXTN= 000002          ST.XLT  000014      006
BIT11 = 004000          B.FMHR  000172          010 FN.EMB  000014      011 SAVESP  000042R         SU.DBU= 000004
BIT12 = 010000          B.FQLS  000162          010 FN.EMC  000016      011 SKELTN  000112R         SU.DQN= 000006
BIT13 = 020000          B.FSAZ  000100          010 FN.FSA  000000      011 SKLNGT= 000040          SU.IDL= 000000
BIT14 = 040000          B.FSBZ  000102          010 FN.FSB  000002      011 SKLSHT= 000033          SU.LOD= 000001
BIT15 = 100000          B.FSCZ  000104          010 FN.FSC  000004      011 SR.ARE  000114      002 SU.SRC= 000002
BIT2  = 000004          B.HBLK  000120          010 FN.LGQ  000034      011 SR.ARS  000106      002 SU.SRR= 000005
BIT3  = 000010          B.HDOC  000114          010 FN.LGU  000036      011 SR.DAY  000010      002 SU.XPD= 000003
BIT4  = 000020          B.HRLP  000126          010 FN.MFO  000024      011 SR.DLT  000014      002 S.HRL = 000240
BIT5  = 000040          B.HRLR  000122          010 FN.MHR  000010      011 SR.ECB  000047      002 TIEF  = 000001
BIT6  = 000100          B.HRLW  000124          010 FN.NMB  000044      011 SR.ECH  000046      002 TILUN = 000005
BIT7  = 000200          B.NMBR  000052          010 FN.OLS  000006      011 SR.ECL  000050      002 TIOUT   000656R
BIT8  = 000400          B.NQRY  000232          010 FN.QRY  000020      011 SR.FIB  000012      002 WN.NTP  000004      012
BIT9  = 001000          B.QLSZ  000106          010 FN.SF0  000030      011 SR.GRE  000100      002 WN.NXT  000006      012
BLANKS  000330R         B.QMAP  000234          010 FN.SF1  000032      011 SR.GRS  000072      002 WN.ROT  000002      012
BNO     000113R         B.QSPL  000316          010 FN.SHD  000042      011 SR.LEN  000122      002 WN.SIZ  000010      012
BSIXIX  000152R         B.QTTM  000076          010 IOSTAT  000036R         SR.LIN  000066      002 WN.SRC  000000      012
BSTINX  000164R         B.QUQP  000056          010 IO.WLB= ****** GX        SR.LIP  000062      002 WN.TYP  000001      012
BSTPTR= ****** G        B.SFDB  000010          010 IS.SUC= ****** GX        SR.MON  000006      002 WORD0 = 000000
BS.CLS= 000002          B.SIZE  000772          010 M     = 000062          SR.NDC  000042      002 WORD1 = 000002
BS.DBU= 000004          B.SNDP  000012          010 MSG1    000044R         SR.NDS  000036      002 WORD2 = 000004
BS.INA= 000000          B.SSQ   000004          010 MSG1L = 000043          SR.NIN  000030      002 WORD3 = 000006
BS.OPN= 000001          B.SSQF  000050          010 MSG1T   000046R         SR.NIP  000022      002 WORD4 = 000010
BS.SRC= 000003          B.STAT  000044          010 N     = 000002          SR.SDB  000032      002 WORD5 = 000012
BS1     000202R         B.STTE  000053          010 NMBQTX  000544R         SR.SRC  000002      002 WORD6 = 000014
BS2     000213R         B.UDOC  000110          010 NQR     000147R         SR.SUN  000000      002 WORD7 = 000016
BS3     000227R         CF.B0 = 000070              NXTBTC  000416R         SR.TWS  000056      002 WORD8 = 000020
BS4     000246R         CF.B2 = 000067              N.BFAC= 000004          SR.WSL  000052      002 WORD9 = 000022
BS5     000255R         CF.B4 = 000066              N.BHGH= 000006          SR.YR   000004      002 WRDVAL= 000024
BS6     000264R         CF.B6 = 000065              N.BTCH= 000004          SR.1IN  000024      002 XBATCH= 000013
BS7     000311R         CF.DR0= 000064              N.BUFB= 004000          SR.1IP  000016      002 XDBLOA= 000004
BYTE0 = 000000          CF.DR1= 000063              N.BUFW= 002000          SS.FID  000002      004 XDBPRO= 000012
BYTE1 = 000001          DBSLEN= 000116              N.FOS = 000764          SS.FNB  000010      004 XDMCIN= 000006
BYTE2 = 000002          DH.BF0  000002          005 N.PKSZ= 000020          SS.FVR  000006      004 XFOSMR= 000007
BYTE3 = 000003          DH.BF1  000004          005 N.PKTS= 000043          SS.LEN  000012      004 XGTSRE= 000014
BYTE4 = 000004          DH.CTL  000000          005 N.QURY= 000031          SS.STT  000000      004 XHITSK= 000011
BYTE5 = 000005          DH.DMC  000010          005 N.SUNT= 000002          START   000354R         XHLMER= 000002
BYTE6 = 000006          DH.FLG  000006          005 PRCERR  000740R         STATE   000121R         XHOTSK= 000010
BYTE7 = 000007          DN.DCK  000000          013 QE.ROI= 000144          STATXT  000524R         XMSCHE= 000000
BYTE8 = 000010          DN.NTP  000004          013 Q.FDSC  000004      007 ST.ASZ  000020      006 XQTS  = 000003
BYTE9 = 000011          DN.NXT  000006          013 Q.NQBK  000000      007 ST.BSZ  000024      006 XQT0  = 000001
BYTVAL= 000012          DN.ROT  000002          013 Q.NUHL  000002      007 ST.BTC  000000      006 XSULOA= 000005
B.BSTA  000054      010 DN.SIZ  000010          013 Q.SIZE  000014      007 ST.CSZ  000030      006 $$$   = 000000R     014
B.CNTX  000046      010 FD.FID  000000          003 RCVCHR  000006R         ST.HRL  000010      006 $$$ARG= 000002
B.COUQ  000060      010 FD.FNB  000006          003 RCVCMD  000004R         ST.LEN  000044      006 $$$OST= 000010
B.FEMA  000132      010 FD.FVR  000004          003 RCVDAT  000000R         ST.QRY  000002      006 $$$T1 = 000000
B.FEMB  000142      010 FD.LEN  000010          003

. ABS.  000000      000
        000744      001
SRCOFF  000122      002
FDSCOF  000010      003
SUSOFF  000012      004
DHROFF  000012      005
```

```
STTOFF  000044     006
QSPLOF  000014     007
BSTOFF  000772     010
FNOFFS  000044     011
WNODOF  000010     012
DNODOF  000010     013
$DPB$$  000010     014
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3639 WORDS  ( 15 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:23
BATCH,BATCH/-SP=[20,1]P,M,BATCH
```

DBSTAT· M[redacted]D·M1110 27-MAR-80 13:38
TABLE·OF·CONTENTS·

10- 2· DISPLAY·DATA·BASE·STATISTICS·

```
                                        .TITLE  DBSTAT
                                        .SBTTL  DISPLAY DATA BASE STATISTICS
                                ;
                                ;
                                        .MCALL  RCVX$C,EXIT$S,QIOW$S
                                ;
                                ;
                                        .GLOBL  SRECPT,$DAT,$CBTA,$CDDMG
                                ;
                                ;
000005                          TILUN=5
000001                          TIEF=1
                                ;
```

```
                                        ;
                                        ; DATA·
                                        ;
000000                                  RCVDAT: .BLKW·  2·      ;RECEIVE·PACKET·
000004                                  RCVCMD: .BLKW·  1       ;COMMAND·CODE·
000006                                  RCVCHR: .BLKW·  1       ;#·OF·PARAMETER·CHARACTERS
000010                                  RCVPRM: .BLKB·  22.     ;PARAMETER·TEXT·
                                        ;
000036                                  IOSTAT: .BLKW·  2·
000042·                                 SAVESP: .BLKW·  1
                                        ;
000044                                  CNTV:   .BLKW·  2·      ;NUMBER·TO·BE·CONVERTED·
                                        ;
                                        ;
                                        ; MESSAGE·SKELETONS·
                                        ;
000050                                  LINE0:
000050      040     040                         .ASCII· /· /
          000002·                       L0=.-LINE0
                                        ;
                                        ;
000052·                                 LINE1:
000052·     125     116     111                 .ASCII· /UNIT:/·
000055      124     072
000057                                  UNB:    .BLKB·  2·
000061      040     040     040                 .ASCII· /·  INIT:/·
000064      111     116     111
000067      124     072
000071                                  DAT:    .BLKB·  9.
000102·     040     040     040                 .ASCII· /·  %·DISK·FULL:/·
000105      045     040     104
000110      111     123     113
000113      040     106     125
000116      114     114     072·
000121                                  PFL:    .BLKB·  3
          000052·                       L1=.-LINE1
                                        ;
                                        ;
000124                                  LINE2:
000124      043     040     117                 .ASCII· /#·OF·SECTOR-·DATA:/·
000127      106     040     123
000132·     105     103     124
000135      117     122     055
000140      040     104     101
000143      124     101     072·
000146                                  DSC:    .BLKB·  9.
000157      040     040     127                 .ASCII· /·  WS·PAST·EOC:/·
000162·     123     040     120
000165      101     123     124
000170      040     105     117
000173      103     072
000175                                  WSE:    .BLKB·  9.
000206      040     040     124                 .ASCII· /·  TOTAL·WS:/·
000211      117     124     101
000214      114     040     127
000217      123     072
000221                                  WST:    .BLKB·  9.
```

```
     53          000106                  L2=.-LINE2
     54                                  ;
     55                                  ;
     56 000232·                          LINE3:
     57 000232·     043     040     117          .ASCII· /#·OF·DOC- INIT:/·
        000235      106     040     104
        000240      117     103     055
        000243      040     111     116
        000246      111     124     072·
     58 000251                          NDI:     .BLKB·  9.
     59 000262·     040     040     040          .ASCII· /·  ID·RANGE:/
        000265      111     104     040
        000270      122·    101     116
        000273      107     105     072·
     60 000276                          LID:     .BLKB·  9.
     61 000307      055                         .ASCII· /-/·
     62·000310                          HID:     .BLKB·  9.
     63          000067                 L3=.-LINE3
     64                                 ;
     65                                 ;
     66 000321                          LINE4:
     67 000321      040     040     040          .ASCII· /·      CURRENT:/·
        000324      040     040     040
        000327      040     103     125
        000332·     122·    122     105
        000335      116     124     072·
     68 000340                          NDC:     .BLKB·  9.
     69 000351      040     040     040          .ASCII· /·  ID·RANGE:/
        000354      111     104     040
        000357      122·    101     116
        000362·     107     105     072·
     70 000365                          LCD:     .BLKB·  9.
     71 000376      055                         .ASCII· /-/·
     72·000377                          HCD:     .BLKB·  9.
     73          000067                 L4=.-LINE4
     74                                 ;
     75                                 ;
     76                                         .EVEN·
     77                                 ;
     78                                 ;
     79                                 ; OUTPUT·LINE·INDEX·TABLE·
     80                                 ;
     81 000410  000050'                 OUTBL:·  .WORD·  LINE0
     82·000412·  000002·                         .WORD·  L0
     83 000414  000052'                          .WORD·  LINE1
     84 000416  000052·                          .WORD·  L1
     85 000420  000124'                          .WORD·  LINE2·
     86 000422·  000106                          .WORD·  L2·
     87 000424  000232'                          .WORD·  LINE3
     88 000426  000067                           .WORD·  L3
     89 000430  000321'                          .WORD·  LINE4
     90 000432·  000067                          .WORD·  L4
     91 000434  000000                           .WORD·  0               ;END·OF·TABLE·
     92·                                ;
     93                                 ;
     94                                 ; MESSAGE·SKELETON·PORTIONS·TO·BE·BLANKED·
     95 000436  000057' 000002          BLNKTB: .WORD·  UNB,2·
```

```
     96 000442  000071' 000011                .WORD   DAT,9.
     97 000446  00012i' 000003                .WORD   PFL,3
     98 000452  000146' 000011                .WORD   DSC,9.
     99 000456  000175' 000011                .WORD   WSE,9.
    100 000462  000221' 000011                .WORD   WST,9.
    101 000466  000251' 000011                .WORD   NDI,9.
    102 000472  000276' 000011                .WORD   LID,9.
    103 000476  000310' 000011                .WORD   HID,9.
    104 000502  000340' 000011                .WORD   NDC,9.
    105 000506  000365' 000011                .WORD   LCD,9.
    106 000512  000377' 000011                .WORD   HCD,9.
    107 000516  000000                        .WORD   0       ;END OF TABLE
    108                                 ;
    109                                 ;
```

```
                                        ;
000520                                  START:
000520  010667  177316                          MOV     SP,SAVESP               ;SAVE IDLE TASK STACK POINTER
                                        ;
000524  016706  177312                  RSTART: MOV     SAVESP,SP               ;RESTORE STACK
000530                                          RCVX$C  ,RCVDAT                 ;TRY FOR PACKET
000536  103003                                  BCC     1$
000540                                          EXIT$S
                                        ;
000546  005004                          1$:     CLR     R4                      ;SREC INDEX
                                        ;
                                        ; NEXT STATUS RECORD
000550  016405  000000G                 NXTSRC: MOV     SRECPT(R4),R5           ;SREC ADDRESS
                                        ; BLANK PORTIONS OF MESSAGE SKELETON
000554  012703  000436'                          MOV     #BLNKTB,R3              ;TABLE OF AREA POINTERS
000560  012301                          1$:     MOV     (R3)+,R1                ;ADDRESS OF AREA TO BE BLANKED
000562  001405                                  BEQ     FRSTLN                  ;DONE WITH BLANKING
000564  012300                                  MOV     (R3)+,R0                ;LENGTH OF AREA
000566  112721  000040                  2$:     MOVB    #' ,(R1)+               ;BLANK IT
000572  077003                                  SOB     R0,2$
000574  000771                                  BR      1$                      ;DO NEXT AREA
                                        ;
                                        ; FIRST LINE
000576  010401                          FRSTLN: MOV     R4,R1                   ;STATUS RECORD NUMBER
000600  006201                                  ASR     R1
000602  005000                                  CLR     R0
000604  071027  000012                          DIV     #10.,R0
000610  062701  000060                          ADD     #'0,R1
000614  110167  177240                          MOVB    R1,UNB+1
000620  062700  000060                          ADD     #'0,R0
000624  110067  177227                          MOVB    R0,UNB
000630  010501                                  MOV     R5,R1                   ;DATE OF INIT
000632  062701  000004                          ADD     #SR.YR,R1
000636  012700  000071'                          MOV     #DAT,R0
000642                                          CALL    $DAT
000646  116500  000046                          MOVB    SR.ECH(R5),R0           ;CALCULATE #OF SECTORS
000652  016501  000050                          MOV     SR.ECL(R5),R1           ;  USED FOR DATA
000656  166501  000034                          SUB     SR.SDB+2(R5),R1
000662  005600                                  SBC     R0
000664  166500  000032                          SUB     SR.SDB(R5),R0
000670  010067  177150                          MOV     R0,CNTV                 ;SAVE #OF SECTORS USED
000674  010167  177146                          MOV     R1,CNTV+2
000700  073027  177774                          ASHC    #-4,R0                  ;SCALE #OF SECTORS USED
000704  010100                                  MOV     R1,R0                   ;SINGLE PRECISION
000706  070027  000144                          MUL     #100.,R0                ;PREPARE FOR %
000712  012702  000007                          MOV     #7,R2                   ;MAX #OF SECTORS ON DISK
000716  012703  105000                          MOV     #105000,R3
000722  166503  000034                          SUB     SR.SDB+2(R5),R3         ;CALCULATE MAX #OF
000726  005602                                  SBC     R2                      ;  USABLE SECTORS
000730  166502  000032                          SUB     SR.SDB(R5),R2
000734  073227  177774                          ASHC    #-4,R2                  ;SCALE MAX #OF SECTORS
000740  071003                                  DIV     R3,R0                   ;% USED
000742  005701                                  TST     R1                      ;ROUND UP
000744  001401                                  BEQ     1$
000746  005200                                  INC     R0
000750  010001                          1$:     MOV     R0,R1                   ;CONVERT TO ASCII
000752  012700  000121'                          MOV     #PFL,R0
```

```
                                        MOV     #10.!BIT9!BIT10!14000,R2
                                        CALL    $CBTA
                                ;
                                ; SECOND LINE
                                        MOV     #DSC,R0                 ;DATA SECTORS USED
                                        MOV     #CNTV,R1
                                        CLR     R2
                                        CALL    $CDDMG
                                        MOV     #WSE,R0                 ;WS AFTER EOC
                                        MOV     SR:WSL(R5),R2           ;CONVERT TO SECTOR COUNT
                                        MOV     SR:WSL+2(R5),R3
                                        ASHC    #-9.,R2
                                        MOV     R2,CNTV
                                        MOV     R3,CNTV+2
                                        MOV     #CNTV,R1
                                        CLR     R2
                                        CALL    $CDDMG
                                        MOV     #WST,R0                 ;TOTAL WS
                                        MOV     SR:TWS(R5),R2           ;CONVERT TO SECTOR COUNT
                                        MOV     SR:TWS+2(R5),R3
                                        ASHC    #-9.,R2
                                        MOV     R2,CNTV
                                        MOV     R3,CNTV+2
                                        MOV     #CNTV,R1
                                        CLR     R2
                                        CALL    $CDDMG
                                ;
                                ; THIRD LINE
                                        MOV     #NDI,R0                 ;# OF DOC AT INIT
                                        MOV     #SR:NDS,R1
                                        CALL    DODBLC
                                        MOV     #LID,R0                 ;LOW GIVEN DOC #
                                        MOV     #SR:GRS+2,R1
                                        CALL    DODBLC
                                        MOV     #HID,R0                 ;HIGH GIVEN DOC #
                                        MOV     #SR:GRE+2,R1
                                        CALL    DODBLC
                                ;
                                ; FOURTH LINE
                                        MOV     #NDC,R0                 ;CURRENT # OF DOC
                                        MOV     #SR:NDC,R1
                                        CALL    DODBLC
                                        MOV     #LCD,R0                 ;LOW CURRENT DOC #
                                        MOV     #SR:ARS+2,R1
                                        CALL    DODBLC
                                        MOV     #HCD,R0                 ;HIGH CURRENT DOC #
                                        MOV     #SR:ARE+2,R1
                                        CALL    DODBLC
                                ;
                                ; OUTPUT DATA LINES
                                        MOV     #OUTBL,R2               ;TABLE POINTER
                                OUTLOP: MOV     (R2)+,R0                ;DATA ADDRESS
                                        BEQ     1$                      ;DONE
                                        MOV     (R2)+,R1                ;DATA LENGTH
                                        CALL    TIOUT                   ;OUTPUT
                                        BR      OUTLOP
                                ;
```

```
225 001240  062704  000002          1$:     ADD     #2,R4                   ;NEXT SU INDEX
226 001244  020427  000004                  CMP     R4,#N.SUNT*2            ;BRANCH IF DONE
227 001250  103002                          BHIS    2$
228 001252  000167  177272                  JMP     NXTSRC                  ;DO NEXT SU
229                                         ;
230 001256  000167  177242          2$:     JMP     RSTART                  ;NEXT PACKET
231                                         ;
```

```
                                        ;
                                        ; DO DOUBLE PRECISION CONVERSION
                                        ;
                                        ; INPUT - R0: OUTPUT DATA ADDRESS
                                        ;         R1: STATUS RECORD INDEX
                                        ;         R5: STATUS RECORD ADDRESS
                                        ;
001262  060501                          DODBLC: ADD     R5,R1                   ;BINARY DATA ADDRESS
001264  005002                                  CLR     R2
001266                                          CALL    $CDDMG                  ;CONVERT
001272                                          RTN
                                        ;
                                        ;
                                        ;
                                        ; OUTPUT A BATCH STATUS LINE
                                        ;
                                        ; INPUT - R0: DATA START ADDRESS
                                        ;         R1: LENGTH OF TEXT
                                        ;
001274                                  TIOUT:  QIOW$S  #IO.WLB,#TILUN,#TIEF,,#IOSTAT,,<R0,R1,#40>
001344  122767  000000G 176464                 CMPB    #IS.SUC,IOSTAT
001352  001001                                  BNE     PRCERR
001354                                          RTN
                                        ;
                                        ; ERROR
                                        ;
001356                                  PRCERR:
001356  000167  177142                          JMP     RSTART
                                        ;
                                        ;
000520'                                         .END    START
```

```
BITVAL= 000000          B.HRLW  000124      010 FN.MFO  000024      011 RCVCMD  000004R         SU.DBU= 000004
BIT0  = 000001          B.NMBR  000052      010 FN.MHR  000010      011 RCVDAT  000000R         SU.DON= 000006
BIT1  = 000002          B.NQRY  000232      010 FN.NMB  000044      011 RCVPRM  000010R         SU.IDL= 000000
BIT10 = 002000          B.QLSZ  000106      010 FN.QLS  000006      011 RSTART  000524R         SU.LOD= 000001
BIT11 = 004000          B.QMAP  000234      010 FN.QRY  000020      011 R.VXBA= 000006          SU.SRC= 000002
BIT12 = 010000          B.QSPL  000316      010 FN.SF0  000030      011 R.VXTN= 000002          SU.SRR= 000005
BIT13 = 020000          B.QTTM  000076      010 FN.SF1  000032      011 SAVESP  000042R         SU.XPD= 000003
BIT14 = 040000          B.QUQP  000056      010 FN.SHD  000042      011 SRECPT= ****** G        S.HRL = 000240
BIT15 = 100000          B.SFDB  000010      010 FRSTLN  000576R         SR.ARE  000114      002 TIEF  = 000001
BIT2  = 000004          B.SIZE  000772      010 HCD     000377R         SR.ARS  000106      002 TILUN = 000005
BIT3  = 000010          B.SNDP  000012      010 HID     000310R         SR.DAY  000010      002 TIOUT   001274R
BIT4  = 000020          B.SSQ   000004      010 IOSTAT  000036R         SR.DLT  000014      002 UNB     000057R
BIT5  = 000040          B.SSQF  000050      010 IO.WLB= ****** GX       SR.ECB  000047      002 WN.NTP  000004      012
BIT6  = 000100          B.STAT  000044      010 IS.SUC= ****** GX       SR.ECH  000046      002 WN.NXT  000006      012
BIT7  = 000200          B.STTE  000053      010 LCD     000365R         SR.ECL  000050      002 WN.ROT  000002      012
BIT8  = 000400          B.UDOC  000110      010 LID     000276R         SR.FIB  000012      002 WN.SIZ  000010      012
BIT9  = 001000          CF.B0 = 000070          LINE0   000050R         SR.GRE  000100      002 WN.SRC  000000      012
BLNKTB  000436R         CF.B2 = 000067          LINE1   000052R         SR.GRS  000072      002 WN.TYP  000001      012
BS.CLS= 000002          CF.B4 = 000066          LINE2   000124R         SR.LEN  000122      002 WORD0 = 000000
BS.DBU= 000004          CF.B6 = 000065          LINE3   000232R         SR.LIN  000066      002 WORD1 = 000002
BS.INA= 000000          CF.DR0= 000064          LINE4   000321R         SR.LIP  000062      002 WORD2 = 000004
BS.OPN= 000001          CF.DR1= 000063          L0    = 000002          SR.MON  000006      002 WORD3 = 000006
BS.SRC= 000003          CNTV    000044R         L1    = 000052          SR.NDC  000042      002 WORD4 = 000010
BYTE0 = 000000          DAT     000071R         L2    = 000106          SR.NDS  000036      002 WORD5 = 000012
BYTE1 = 000001          DBSLEN= 000116          L3    = 000067          SR.NIN  000030      002 WORD6 = 000014
BYTE2 = 000002          DH.BF0  000002      005 L4    = 000067          SR.NIP  000022      002 WORD7 = 000016
BYTE3 = 000003          DH.BF1  000004      005 M     = 000062          SR.SDB  000032      002 WORD8 = 000020
BYTE4 = 000004          DH.CTL  000000      005 N     = 000002          SR.SRC  000002      002 WORD9 = 000022
BYTE5 = 000005          DH.DMC  000010      005 NDC     000340R         SR.SUN  000000      002 WRDVAL= 000024
BYTE6 = 000006          DH.FLG  000006      005 NDI     000251R         SR.TWS  000056      002 WSE     000175R
BYTE7 = 000007          DN.DCK  000000      013 NXTSRC  000550R         SR.WSL  000052      002 WST     000221R
BYTE8 = 000010          DN.NTP  000004      013 N.BFAC= 000004          SR.YR   000004      002 XBATCH= 000013
BYTE9 = 000011          DN.NXT  000006      013 N.BHGH= 000006          SR.1IN  000024      002 XDBLOA= 000004
BYTVAL= 000012          DN.ROT  000002      013 N.BTCH= 000004          SR.1IP  000016      002 XDBPRO= 000012
B.BSTA  000054      010 DN.SIZ  000010      013 N.BUFB= 004000          SS.FID  000002      004 XDMCIN= 000006
B.CNTX  000046      010 DODBLC  001262R         N.BUFW= 002000          SS.FNB  000010      004 XFOSMR= 000007
B.COUQ  000060      010 DSC     000146R         N.FOS = 000764          SS.FVR  000006      004 XGTSRE= 000014
B.FEMA  000132      010 FD.FID  000000      003 N.PKSZ= 000020          SS.LEN  000012      004 XHITSK= 000011
B.FEMB  000142      010 FD.FNB  000006      003 N.PKTS= 000043          SS.STT  000000      004 XHLMER= 000002
B.FEMC  000152      010 FD.FVR  000004      003 N.QURY= 000031          START   000520R         XHOTSK= 000010
B.FFSA  000202      010 FD.LEN  000010      003 N.SUNT= 000002          ST.ASZ  000020      006 XMSCHE= 000000
B.FFSB  000212      010 FN.DBR  000026      011 OUTBL   000410R         ST.BSZ  000024      006 XQTS  = 000003
B.FFSC  000222      010 FN.DBS  000022      011 OUTLOP  001224R         ST.BTC  000000      006 XQT0  = 000001
B.FMHR  000172      010 FN.DHR  000040      011 PFL     000121R         ST.CSZ  000030      006 XSULOA= 000005
B.FQLS  000162      010 FN.EMA  000012      011 PRCERR  001356R         ST.HRL  000010      006 $CBTA = ****** G
B.FSAZ  000100      010 FN.EMB  000014      011 QE.RQ1= 000144          ST.LEN  000044      006 $CDDMG= ****** G
B.FSBZ  000102      010 FN.EMC  000016      011 Q.FDSC  000004      007 ST.QRY  000002      006 $DAT  = ****** G
B.FSCZ  000104      010 FN.FSA  000000      011 Q.NQBK  000000      007 ST.QSZ  000034      006 $$$   = 000000R     014
B.HBLK  000120      010 FN.FSB  000002      011 Q.NUHL  000002      007 ST.SCH  000040      006 $$$ARG= 000002
B.HDOC  000114      010 FN.FSC  000004      011 Q.SIZE  000014      007 ST.UHL  000004      006 $$$OST= 000010
B.HRLP  000126      010 FN.LGQ  000034      011 RCVCHR  000006R         ST.XLT  000014      006 $$$T1 = 000000
B.HRLR  000122      010 FN.LGU  000036      011

. ABS.  000000      000
        001362      001
SRCOFF  000122      002
FDSCOF  000010      003
```

```
DBSTAT  M[illegible]0  M1110  27-MAR-80 13:38  PAGE 13-2
SYMBOL TA[illegible]E

SUSOFF  000012    004
DHROFF  000012    005
STTOFF  000044    006
QSPLOF  000014    007
BSTOFF  000772    010
FNOFFS  000044    011
WNODOF  000010    012
DNODOF  000010    013
$DPB$$  000010    014
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3607 WORDS  ( 15 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:26
DBSTAT,DBSTAT/-SP=[20,1]DP,M,DBSTAT
```

10- 2. COLLECT HOURLY STATISTICS.

```
                                        .TITLE..HRSTAT.
                                        .SBTTL. COLLECT.HOURLY STATISTICS.
                                ;
                                ;
                                        .MCALL. GTIM$C,EXIT$S,ROST$,SDAT$
                                ;
                                ;
                                        .GLOBL. HRSTFG,LDAY,LHOUR.
                                        .GLOBL. CHSTAT,CDSTAT,LHSTAT,LDSTAT.
                                ;
                                ;
                                ; DATA.
                                ;
000000                          TIMBUF:..BLKW.  8.      ;CURRENT.TIME.
000020                          SNDPKT: .BLKW.  13.
000052. 000004                  CMD:    .WORD.  4
000054  074741  077770                  .RAD50  /STATS/.
                                ;
```

```
                                        ;
000060                                  START:
000060                                          GTIM$C  TIMBUF                  ;CURRENT TIME
                                        ;
000066  005767  000000G                         TST     HRSTFG                  ;BRANCH IF NOT
000072  001010                                  BNE     1$                      ;  FIRST TIME THROUGH
000074  016767  177704  000000G                 MOV     TIMBUF+G,TIDA,LDAY      ;INIT TIME
000102  016767  177700  000000G                 MOV     TIMBUF+G,TIHR,LHOUR
000110  005267  000000G                         INC     HRSTFG                  ;SHOW INIT
                                        ;
000114  026767  177666  000000G 1$:      CMP     TIMBUF+G,TIHR,LHOUR     ;BRANCH IF HOUR TURNED OVER
000122  001003                                  BNE     HTURN
000124                                          EXIT$S                  ;ELSE NOTHING TO DO
                                        ;
                                        ; HOUR TURNED OVER
000132  016767  177650  000000G HTURN:  MOV     TIMBUF+G,TIHR,LHOUR     ;NEW HOUR
                                        ; ACCUMULATE HOURLY STATS
000140  012700  000000G                         MOV     #CHSTAT,R0
000144  012701  000000G                         MOV     #LHSTAT,R1
000150  012702  000000G                         MOV     #CDSTAT,R2
000154  061022                                  ADD     (R0),(R2)+              ;# OF BATCHES
000156  011021                                  MOV     (R0),(R1)+
000160  005020                                  CLR     (R0)+
000162  061022                                  ADD     (R0),(R2)+              ;# OF QUERIES
000164  011021                                  MOV     (R0),(R1)+
000166  005020                                  CLR     (R0)+
                                        ; ACCUMULATE DOUBLE-WORD COUNTS IN REMAINDER OF STATS RECORD
000170  012703  000010                         MOV     #ST.LEN/4-1,R3          ;# OF DOUBLE-WORD ENTRIES
000174  066062  000002  000002  1$:      ADD     2(R0),2(R2)
000202  005512                                  ADC     (R2)
000204  061012                                  ADD     (R0),(R2)
000206  062702  000004                         ADD     #4,R2
000212  011021                                  MOV     (R0),(R1)+
000214  005020                                  CLR     (R0)+
000216  011021                                  MOV     (R0),(R1)+
000220  005020                                  CLR     (R0)+
000222  077314                                  SOB     R3,1$                   ;DO ALL ENTRIES
                                        ;
000224  026767  177554  000000G                 CMP     TIMBUF+G,TIDA,LDAY      ;BRANCH IF DAY TURNED OVER
000232  001027                                  BNE     DTURN
                                        ;
                                        ;       PRINT HOURLY STATS ON CONSOLE
                                        ;
000234  012704  000052'                         MOV     #CMD,R4
000240  012700  000020'                         MOV     #SNDPKT,R0
000244  012420                                  MOV     (R4)+,(R0)+
000246  005001                                  CLR     R1
000250  010120                                  MOV     R1,(R0)+
000252  000000                                  HALT
000254                                          SDAT$   R4,SNDPKT
000266                                          RQST$   R4
000304                                          EXIT$S                  ;ELSE DONE
                                        ;
                                        ; DAY TURNED OVER
000312  016767  177466  000000G DTURN:  MOV     TIMBUF+G,TIDA,LDAY      ;NEW DAY
                                        ; ACCUMULATE DAILY STATS
000320  012700  000000G                         MOV     #CDSTAT,R0
```

```
     77 000324  012701  000000G                MOV     #LDSTAT,R1
     78 000330  012702  000022                 MOV     #ST.LEN/2,R2            ;SIZE OF STAT AREA
     79 000334  011021                  1$:    MOV     (R0),(R1)+
     80 000336  005020                         CLR     (R0)+
     81 000340  077203                         SOB     R2,1$
     82                                 ;
     83 000342                                 EXIT$S
     84                                 ;
     85                                 ;
     86                                 ;
     87         000060'                        .END    START
```

```
BITVAL=·000000          B.HDOC  000114      010 FN.EMB· 000014      011 R.QSGC=·000015          ST.UHL· 000004      006
BIT0  =·000001          B.HRLP  000126      010 FN.EMC· 000016      011 R.QSPC=·000014          ST.XLT· 000014      006
BIT1  =·000002·         B.HRLR  000122·     010 FN.FSA· 000000      011 R.QSPN=·000006          SU.DBU=·000004
BIT10 =·002000          B.HRLW  000124      010 FN.FSB· 000002·     011 R.QSPR=·000012·         SU.DON=·000006
BIT11 =·004000          B.NMBR  000052·     010 FN.FSC· 000004      011 R.QSTN=·000002·         SU.IDL=·000000
BIT12·=·010000          B.NQRY  000232·     010 FN.LGD· 000034      011 SNDPKT· 000020R·        SU.LOD=·000001
BIT13 =·020000          B.QLSZ  000106      010 FN.LGU· 000036      011 SR.ARE· 000114      002·SU.SRC=·000002
BIT14 =·040000          B.QMAP  000234      010 FN.MFO· 000024      011 SR.ARS· 000106      002·SU.SRR=·000005
BIT15 =·100000          B.QSPL  000316      010 FN.MHR· 000010      011 SR.DAY· 000010      002·SU.XPD=·000003
BIT2· =·000004          B.QTTM  000076      010 FN.NMB· 000044      011 SR.DLT· 000014      002·S.DABA=·000006
BIT3  =·000010          B.QUQP  000856      010 FN.QLS· 000006      011 SR.ECB· 000047      002·S.DAEF=·000010
BIT4  =·000020          B.SFDB  000010      010 FN.QRY· 000020      011 SR.ECH· 000046      002·S.DATN=·000002
BIT5  =·000040          B.SIZE  000772·     010 FN.SF0  000030      011 SR.ECL· 000050      002·S.HRL·=·000240
BIT6  =·000100          B.SNDP  000012·     010 FN.SF1  000032·     011 SR.FIB· 000012·     002·TIMBUF· 000000R·
BIT7  =·000200          B.SSQ·  000004      010 FN.SHD· 000042·     011 SR.GRE· 000100      002·WN.NTP· 000004      012·
BIT8·=·000400           B.SSQF  000050      010 G.TIBA=·000002·         SR.GRS· 000072·     002·WN.NXT· 000006      012·
BIT9  =·001000          B.STAT  000044      010 G.TICP=·000016          SR.LEN· 000122·     002·WN.ROT· 000002      012·
BS.CLS=·000002·         B.STTE  000053      010 G.TICT=·000014          SR.LIN· 000066      002·WN.SIZ· 000010      012·
BS.DBU=·000004          B.UDOC  000110      010 G.TIDA=·000004          SR.LIP· 000062·     002·WN.SRC· 000000      012·
BS.INA=·000000          CDSTAT=·******·G        G.TIHR=·000006          SR.MON· 000006      002·WN.TYP· 000001      012·
BS.OPN=·000001          CF.B0 =·000070          G.TIMI=·000010          SR.NDC· 000042·     002·WORD0 =·000000
BS.SRC=·000003          CF.B2·=·000067          G.TIMO=·000002·         SR.NDS· 000036      002·WORD1 =·000002
BYTE0 =·000000          CF.B4 =·000066          G.TISC=·000012·         SR.NIN· 000030      002·WORD2·=·000004
BYTE1 =·000001          CF.B6 =·000065          G.TIYR=·000000          SR.NIP· 000022·     002·WORD3 =·000006
BYTE2·=·000002·         CF.DR0=·000064          HRSTFG=·******·G·       SR.SDB· 000032·     002·WORD4 =·000010
BYTE3 =·000003          CF.DR1=·000063          HTURN·  000132R·        SR.SRC· 000002·     002·WORD5 =·000012
BYTE4 =·000004          CHSTAT=·******·G        LDAY·  =·******·G·      SR.SUN· 000000      002·WORD6 =·000014
BYTE5 =·000005          CMD·    000052R·        LDSTAT=·******·G·       SR.TWS· 000056      002·WORD7 =·000016
BYTE6 =·000006          DBSLEN=·000116          LHOUR·=·******·G·       SR.WSL· 000052·     002·WORD8 =·000020
BYTE7 =·000007          DH.BF0  000002·     005 LHSTAT=·******·G·       SR.YR·  000004      002·WORD9 =·000022
BYTE8·=·000010          DH.BF1  000004      005 M·     =·000062·        SR.1IN· 000024      002·WRDVAL=·000024
BYTE9 =·000011          DH.CTL  000000      005 N·     =·000002·        SR.1IP· 000016      002·XBATCH=·000013
BYTVAL=·000012·         DH.DMC  000010      005 N.BFAC=·000004          SS.FID· 000002·     004 XDBLOA=·000004
B.BSTA· 000054      010 DH.FLG  000006      005 N.BHGH=·000006          SS.FNB· 000010      004 XDBPRO=·000012
B.CNTX· 000046      010 DN.DCK  000000      013 N.BTCH=·000004          SS.FVR· 000006      004 XDMCIN=·000006
B.COUQ· 000060      010 DN.NTP  000004      013 N.BUFB=·004000          SS.LEN· 000012·     004 XFOSMR=·000007
B.FEMA· 000132·     010 DN.NXT  000006      013 N.BUFW=·002000          SS.STT· 000000      004 XGTSRE=·000014
B.FEMB· 000142·     010 DN.ROT  000002·     013 N.FQS·=·000764          START·  000060R·        XHITSK=·000011
B.FEMC· 000152·     010 DN.SIZ  000010      013 N.PKSZ=·000020          ST.ASZ· 000020      006 XHLMER=·000002
B.FFSA· 000202·     010 DTURN·  000312R·        N.PKTS=·000043          ST.BSZ· 000024      006 XHOTSK=·000010
B.FFSB· 000212·     010 FD.FID  000000      003 N.QURY=·000031          ST.BTC· 000000      006 XMSCHE=·000000
B.FFSC· 000222·     010 FD.FNB  000006      003 N.SUNT=·000002·         ST.CSZ· 000030      006 XQTS  =·000003
B.FMHR· 000172·     010 FD.FVR  000004      003 QE.RQ1=·000144          ST.HRL· 000010      006 XQT0  =·000001
B.FQLS· 000162·     010 FD.LEN  000010      003 Q.FDSC· 000004      007 ST.LEN· 000044      006 XSULOA=·000005
B.FSAZ· 000100      010 FN.DBR  000026      011 Q.NQBK· 000000      007 ST.QRY· 000002·     006 $$$   =·000000R·      014
B.FSBZ· 000102·     010 FN.DBS  000022·     011 Q.NUHL· 000002·     007 ST.QSZ· 000034      006 $$$OST=·000016
B.FSCZ· 000104      010 FN.DHR  000040      011 Q.SIZE· 000014      007 ST.SCH· 000040      006 $$$T1 =·000002
B.HBLK· 000120      010 FN.EMA  000012·     011

. ABS.  000000  000
        000350  001
SRCOFF· 000122· 002·
FDSCOF· 000010  003
SUSOFF· 000012· 004
DHROFF· 000012· 005
STTOFF· 000044  006
QSPLOF· 000014  007
```

```
HRSTAT  MACRO M1110  27-MAR-80 13:38  PAGE 11-3
SYMBOL TABLE

BSTOFF  000772     010
FNOFFS  000044     011
WNODOF  000010     012
DNODOF  000010     013
$DPB$$  000004     014
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3092 WORDS  ( 13 PAGES)
DYNAMIC MEMORY:  3860 WORDS  ( 14 PAGES)
ELAPSED TIME:  00:00:20
MKSTAT,HRSTAT/-SP=[20,1]DP,M,HRSTAT
```

10- 2 DISPLAY SYSTEM STATISTICS

```
                                        .TITLE  STATS
                                        .SBTTL  DISPLAY SYSTEM STATISTICS
                                ;
                                ;
                                        .MCALL  RCVX$C,QIOW$S
                                ;
                                ;
                                        .GLOBL  CHSTAT,LHSTAT,CDSTAT,LDSTAT
                                        .GLOBL  $CBTA,$DDIV,$CDDMG
                                ;
                                ;
000005                          TILUN=5
000001                          TIEF=1
027012                          CNVT5=10.!BIT9!BIT10!24000
017012                          CNVT3=10.!BIT9!BIT10!14000
011012                          CNVT2=10.!BIT9!10000
013012                          CNVT2S=10.!BIT9!BIT10!10000
                                ;
```

```
                                        ;
                                        ; DATA
000000                                  RCVDAT: .BLKW  2        ;RECEIVE PACKET
000004                                  RCVCMD: .BLKW  1        ;COMMAND CODE
000006                                  RCVCHR: .BLKW  1        ;# OF PARAM CHARACTERS
000010                                  RCVPRM: .BLKB  22.      ;PARAM TEXT
                                        ;
000036                                  IOSTAT: .BLKW  2
                                        ;
000042                                  DBLPVL: .BLKW  2        ;DOUBLE PRECISION VALUE TO BE CONVERTED
                                        ;
                                        ;
                                        ; MESSAGE SKELETONS
                                        ;
000046                                  LINE1:
000046  102  101  124                          .ASCII  /BATCHES:/
000051  103  110  105
000054  123  072
000056                                  NBT:    .BLKB  5
000063  040  040  040                          .ASCII  /      /
000066  040  040  040
000071  040
000072                                  TST:    .BLKB  12.
        000040                          SL1=.-LINE1
000106  040  101  126                          .ASCII  /AVERAGE PER BATCH/
000111  105  122  101
000114  107  105  040
000117  120  105  122
000122  040  102  101
000125  124  103  110
        000062                          L1=.-LINE1
                                        ;
000130                                  LINE2:
000130  040  040                              .ASCII  /  /
        000002                          L2=.-LINE2
                                        ;
000132                                  LINE3:
000132  121  125  105                          .ASCII  *QUERIES  DOC/   DOC IN  XLATE     *
000135  122  111  105
000140  123  040  040
000143  104  117  103
000146  057  040  040
000151  040  104  117
000154  103  040  111
000157  116  040  040
000162  130  114  101
000165  124  105  040
000170  040  040  040
000173  040
000174  106  123  101                          .ASCII  *FSA-A  FSA-B  FSA-C  QLS  SEARCH TIME/*
000177  055  101  040
000202  040  106  123
000205  101  055  102
000210  040  040  106
000213  123  101  055
000216  103  040  040
000221  121  114  123
```

```
        000224     040     040     123
        000227     105     101     122.
        000232.    103     110     040
        000235     124     111     115
        000240     105     057
    50          000110                  L3=.-LINE3
    51                                  ;
    52.000242.                          LINE4:
    53 000242.     040     040     040          .ASCII. *         UHL.    HRL.     TIME.(S)  WORDS. WORDS*
       000245      040     040     040
       000250      040     040     040
       000253      125     110     114
       000256      040     040     040
       000261      040     110     122.
       000264      114     040     040
       000267      040     040     040
       000272.     124     111     115
       000275      105     040     050
       000300      123     051     040
       000303      040     127     117
       000306      122.    104     123
       000311      040     040     127
       000314      117     122     104
       000317      123
    54 000320      040     040     127          .ASCII. *  WORDS. WORDS. SU.(MM:SS)*
       000323      117     122     104
       000326      123     040     040
       000331      127     117     122.
       000334      104     123     040
       000337      040     123     125
       000342.     040     050     115
       000345      115     072     123
       000350      123     051
    55       .   000110                  L4=.-LINE4
    56                                  ;
    57 000352.                          LINE5:
    58 000352.                          QRY:    .BLKB.  9.
    59 000363                           DUH:    .BLKB.  8.
    60 000373                           DHR:    .BLKB.  7
    61 000402.                          XTM:    .BLKB.  10.
    62.000414                           FSA:    .BLKB.  7
    63 000423                           FSB:    .BLKB.  7
    64 000432.                          FSC:    .BLKB.  7
    65 000441                           ILS:    .BLKB.  7
    66 000450                           SRT:    .BLKB.  5
    67          000103                  L5=.-LINE5
    68                                  ;
    69                                          .EVEN.
    70                                  ;
    71                                  ; OUTPUT.LINE.INDEX.TABLE.
    72.                                 ;
    73 000456  000046' 000062           OUTBL:  .WORD.  LINE1,L1
    74 000462. 000130' 000002                   .WORD.  LINE2,L2.
    75 000466  000132' 000110                   .WORD.  LINE3,L3
    76 000472. 000242' 000110                   .WORD.  LINE4,L4
    77 000476  000352' 000103                   .WORD.  LINE5,L5
    78 000502. 000000                           .WORD.  0               ;END.OF.TABLE.
```

```
STATS  MACRO M1110  27-MAR-80 13:39  PAGE 14-2
SYMBOL TABLE

FDSCOF  000010     003
SUSOFF  000012     004
DHROFF  000012     005
STTOFF  000044     006
QSPLOF  000014     007
@STOFF  000772     010
FNOFFS  000044     011
UNODOF  000010     012
DNODOF  000010     013
$DPB$$  000010     014
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3718 WORDS  (15 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:26
STATS,STATS/-SP=[20,1]P,M,STATS
```

```
                                        ;
                                        ; CONVERT DOUBLE PRECISION VALUE TO ASCII
                                        ;
                                        ; R3= ASCII DATA ADDRESS
                                        ; R4= STATS VALUE INDEX
                                        ;
001506  060504                  DBLCVT: ADD     R5,R4                   ;DOUBLE PRECISION DIVIDE
001510  012401                          MOV     (R4)+,R1                ;  BY BATCH NUMBER
001512  011402                          MOV     (R4),R2
001514  016500  000000                  MOV     ST.BTC(R5),R0
001520                                  CALL    $DDIV
001524  010167  176312                  MOV     R1,DBLPVL               ;DOUBLE PRECISION CONVERT
001530  010267  176310                  MOV     R2,DBLPVL+2             ;  TO ASCII
001534  012701  000042'                  MOV     #DBLPVL,R1
001540  010300                          MOV     R3,R0
001542  005002                          CLR     R2
001544                                  CALL    $CDDMG
001550                                  RTN
                                        ;
                                        ;
000564'                                 .END    START
```

```
BITVAL= 000000          B.NQRY  000232          010 FN.EMC  000016      011 Q.FDSC  000004      007 ST.UHL  000004      006
BIT0  = 000001          B.QLSZ  000106          010 FN.FSA  000000      011 Q.NQBK  000000      007 ST.XLT  000014      006
BIT1  = 000002          B.QMAP  000234          010 FN.FSB  000002      011 Q.NUHL  000002      007 SU.DBU= 000004
BIT10 = 002000          B.QSPL  000316          010 FN.FSC  000004      011 Q.SIZE  000014      007 SU.DON= 000006
BIT11 = 004000          B.QTTM  000076          010 FN.LGQ  000034      011 RCVCHR  000006R         SU.IDL= 000000
BIT12 = 010000          B.QUQP  000056          010 FN.LGU  000036      011 RCVCMD  000004R         SU.LOD= 000001
BIT13 = 020000          B.SFDB  000010          010 FN.MFQ  000024      011 RCVDAT  000000R         SU.SRC= 000002
BIT14 = 040000          B.SIZE  000772          010 FN.MHR  000010      011 RCVPRM  000010R         SU.SRR= 000005
BIT15 = 100000          B.SNDP  000012          010 FN.NMB  000044      011 R.VXBA= 000006          SU.XPD= 000003
BIT2  = 000004          B.SSQ   000004          010 FN.QLS  000006      011 R.VXTN= 000002          S.HRL = 000240
BIT3  = 000010          B.SSQF  000050          010 FN.QRY  000020      011 SL1   = 000040          TIEF  = 000001
BIT4  = 000020          B.STAT  000044          010 FN.SF0  000030      011 SRT     000450R         TILUN = 000005
BIT5  = 000040          B.STTE  000053          010 FN.SF1  000032      011 SR.ARE  000114      002 TIOUT   001434R
BIT6  = 000100          B.UDOC  000110          010 FN.SHD  000042      011 SR.ARS  000106      002 TST     000072R
BIT7  = 000200          CDS     000534R             FSA     000414R         SR.DAY  000010      002 WN.NTP  000004      012
BIT8  = 000400          CDSTAT= ****** G            FSB     000423R         SR.DLT  000014      002 WN.NXT  000006      012
BIT9  = 001000          CF.B0 = 000070              FSC     000432R         SR.ECB  000047      002 WN.ROT  000002      012
BS.CLS= 000002          CF.B2 = 000067              IOSTAT  000036R         SR.ECH  000046      002 WN.SIZ  000010      012
BS.DBU= 000004          CF.B4 = 000066              IO.WLB= ****** GX           SR.ECL  000050      002 WN.SRC  000000      012
BS.INA= 000000          CF.B6 = 000065              LDS     000550R         SR.FIB  000012      002 WN.TYP  000001      012
BS.OPN= 000001          CF.DR0= 000064              LDSTAT= ****** G            SR.GRE  000100      002 WORD0 = 000000
BS.SRC= 000003          CF.DR1= 000063              LHS     000520R         SR.GRS  000072      002 WORD1 = 000002
BYTE0 = 000000          CHS     000504R             LHSTAT= ****** G            SR.LEN  000122      002 WORD2 = 000004
BYTE1 = 000001          CHSTAT= ****** G            LINE1   000046R         SR.LIN  000066      002 WORD3 = 000006
BYTE2 = 000002          CNVT2 = 011012              LINE2   000130R         SR.LIP  000062      002 WORD4 = 000010
BYTE3 = 000003          CNVT2S= 013012              LINE3   000132R         SR.MON  000086      002 WORD5 = 000012
BYTE4 = 000004          CNVT3 = 017012              LINE4   000242R         SR.NDC  000042      002 WORD6 = 000014
BYTE5 = 000005          CNVT5 = 027012              LINE5   000352R         SR.NDS  000036      002 WORD7 = 000016
BYTE6 = 000006          DBLCVT  001506R             LOOP5   001270R         SR.NIN  000030      002 WORD8 = 000020
BYTE7 = 000007          DBLPVL  000042R             L1    = 000062          SR.NIP  000022      002 WORD9 = 000022
BYTE8 = 000010          DBSLEN= 000116              L2    = 000002          SR.SDB  000032      002 WRDVAL= 000024
BYTE9 = 000011          DHR     000373R             L3    = 000110          SR.SRC  000002      002 XBATCH= 000013
BYTVAL= 000012          DH.BF0  000002          005 L4    = 000110          SR.SUN  000000      002 XDBLOA= 000004
B.BSTA  000054      010 DH.BFI  000004          005 L5    = 000103          SR.TWS  000056      002 XDBPRO= 000012
B.CNTX  000046      010 DH.CTL  000000          005 M     = 000062          SR.WSL  000052      002 XDMCIN= 000006
B.CQUQ  000060      010 DH.DMC  000010          005 MVTSTG  000712R         SR.YR   000004      002 XFOSMR= 000007
B.FEMA  000132      010 DH.FLG  000006          005 N     = 000002          SR.1IN  000024      002 XGTSRE= 000014
B.FEMB  000142      010 DN.DCK  000000          013 NBT     000056R         SR.1IP  000016      002 XHITSK= 000011
B.FEMC  000152      010 DN.NTP  000004          013 N.BFAC= 000004          SS.FID  000002      004 XHLMER= 000002
B.FFSA  000202      010 DN.NXT  000006          013 N.BHGH= 000006          SS.FNB  000010      004 XHOTSK= 000010
B.FFSB  000212      010 DN.ROT  000002          013 N.BTCH= 000004          SS.FVR  000006      004 XMSCHE= 000000
B.FFSC  000222      010 DN.SIZ  000010          013 N.BUFB= 004000          SS.LEN  000012      004 XQTS  = 000003
B.FMHR  000172      010 DOLIN5  001034R             N.BUFW= 002000          SS.STT  000000      004 XQT0  = 000001
B.FQLS  000162      010 DUH     000363R             N.FOS = 000764      START   000564R             XSULOA= 000005
B.FSAZ  000100      010 FD.FID  000000          003 N.PKSZ= 000020          ST.ASZ  000020      006 XTM     000402R
B.FSBZ  000102      010 FD.FNB  000006          003 N.PKTS= 000043          ST.BSZ  000024      006 $CBTA = ****** G
B.FSCZ  000104      010 FD.FVR  000004          003 N.QURY= 000031          ST.BTC  000000      006 $CDDMG= ****** G
B.HBLK  000120      010 FD.LEN  000010          003 N.SUNT= 000002          ST.CSZ  000030      006 $DDIV = ****** G
B.HDOC  000114      010 FN.DBR  000026          011 OUTBL   000456R         ST.HRL  000010      006 $$$     000000R     014
B.HRLP  000126      010 FN.DBS  000022          011 OUTLOP  001414R         ST.LEN  000044      006 $$$ARG= 000002
B.HRLR  000122      010 FN.DHR  000040          011 QE.RQI= 000144          ST.QRY  000002      006 $$$OST= 000010
B.HRLW  000124      010 FN.EMA  000012          011 QLS     000441R         ST.QSZ  000034      006 $$$T1 = 000000
B.NMBR  000052      010 FN.EMB  000014          011 QRY     000352R         ST.SCH  000040      006

. ABS.  000000      000
        001552      001
SRCOFF  000122      002
```

```
                                        ;
                                        ; LINE 5
                                        ;
                                        ; R5= STATISTICS AREA ADDRESS
                                        ;
001034  012701  000352'         DOLIN5: MOV     #LINE5,R1               ;CLEAR LINE 5 TO BLANKS
001040  012700  000103                  MOV     #L5,R0
001044  112721  000040          1$:     MOVB    #' ,(R1)+
001050  077003                          SOB     R0,1$
                                        ; QUERIES PER BATCH
001052  016503  000002                  MOV     ST.QRY(R5),R3           ;QUERIES
001056  005002                          CLR     R2
001060  071265  000000                  DIV     ST.BTC(R5),R2           ;CALCULATE WHOLE NUMBER
001064  010201                          MOV     R2,R1                   ;FORMAT WHOLE NUMBER
001066  012700  000352'                 MOV     #QRY,R0
001072  012702  017012                  MOV     #CNVT3,R2
001076                                  CALL    $CBTA
001102  112720  000056                  MOVB    #'.,(R0)+               ;DECIMAL POINT
001106  070327  000144                  MUL     #100.,R3                ;CALCULATE FRACTION
001112  005002                          CLR     R2
001114  071265  000000                  DIV     ST.BTC(R5),R2
001120  010201                          MOV     R2,R1                   ;FORMAT FRACTION
001122  012702  011012                  MOV     #CNVT2,R2
001126                                  CALL    $CBTA
                                        ;
001132  012703  000363'                 MOV     #DUH,R3                 ;DOC/UHL PER BATCH
001136  012704  000004                  MOV     #ST.UHL,R4
001142                                  CALL    DBLCVT                  ;CALCULATE AND FORMAT
001146  012703  000373'                 MOV     #DHR,R3                 ;DOC-HRL PER BATCH
001152  012704  000010                  MOV     #ST.HRL,R4
001156                                  CALL    DBLCVT                  ;CALCULATE AND FORMAT
                                        ; CALCULATE TRANSLATE TIME IN SECONDS
001162  016502  000014                  MOV     ST.XLT(R5),R2           ;XLATE TIME IN TICS
001166  016503  000016                  MOV     ST.XLT+2(R5),R3
001172  016501  000000                  MOV     ST.BTC(R5),R1           ;DIVIDE BY NUMBER OF
001176  070127  000074                  MUL     #60.,R1                 ;  BATCHES TIMES TICKS/SEC
001202  071201                          DIV     R1,R2
001204  012700  000402'                 MOV     #XTM,R0                 ;CONVERT SECONDS
001210  010201                          MOV     R2,R1                   ;  TO ASCII
001212  012702  017012                  MOV     #CNVT3,R2
001216                                  CALL    $CBTA
001222  112720  000056                  MOVB    #'.,(R0)+               ;DECIMAL POINT
001226  070327  000144                  MUL     #100.,R3                ;CALCULATE DECIMAL FRACTION
001232  005002                          CLR     R2                      ;  OF SECONDS
001234  071227  000074                  DIV     #60.,R2
001240  010201                          MOV     R2,R1                   ;CONVERT FRACTION TO ASCII
001242  012702  011012                  MOV     #CNVT2,R2
001246                                  CALL    $CBTA
                                        ; CONVERT NEXT 4 VALUES
001252  012746  000004                  MOV     #4.,-(SP)               ;COUNT
001256  010504                          MOV     R5,R4                   ;POINT TO FIRST STATS VALUE
001260  062704  000020                  ADD     #ST.ASZ,R4
001264  012700  000414'                 MOV     #FSA,R0                 ;START OF OUTPUT DATA
001270  012402                  LOOP5:  MOV     (R4)+,R2                ;DIVIDE BY NUMBER OF BATCHES
001272  012403                          MOV     (R4)+,R3
001274  071265  000000                  DIV     ST.BTC(R5),R2
001300  010201                          MOV     R2,R1                   ;CONVERT TO ASCII
```

```
STATS.  M[redacted]0.M1110  27-MAR-80 13:39  PAGE.13-1
DISPLAY.S[redacted]EM.STATISTICS.

     201 001302.  012702.  027012                    MOV.    #CNVT5,R2.
     202.001306                                      CALL.   $CBTA.
     203 001312.  062700   000002                    ADD.    #2,R0                   ;TWO.SPACES.
     204 001316   005316                             DEC.    (SP)                    ;LOOP.4.TIMES.
     205 001320   001363                             BNE.    LOOP5
     206 001322.  005726                             TST.    (SP)+                   ;CLEAN.STACK.
     207                                  ; CALCULATE.SEARCH.TIME.IN.MINUTES.AND.SECONDS.
     208 001324   016502.  000040                    MOV.    ST.SCH(R5),R2.          ;SECONDS.
     209 001330   016503   000042                    MOV.    ST.SCH+2(R5),R3
     210 001334   016501   000000                    MOV.    ST.BTC(R5),R1           ;DIVIDE.BY.NUMBER.OF.BATCHES.
     211 001340   071201                             DIV.    R1,R2.                  ;R2.=.SECONDS.PER.SEARCH.
     212.                                                                            ;R3.=.REMAINDER.
     213 001342.  010203                             MOV.    R2,R3                   ;R3.=.SECONDS.PER.SEARCH.
     214 001344   006203                             ASR.    R3                      ;R3.=.SECONDS.PER.SEARCH.PER.SU.
     215 001346   005002.                            CLR.    R2.
     216 001350   071227   000074                    DIV.    #60.,R2.                ;R2.=.MINUTES.
     217                                                                            ;R3.=.SECONDS.
     218 001354   012700   000450'                   MOV.    #SRT,R0                 ;CONVERT.MINUTES.TO.ASCII.
     219 001360   010201                             MOV.    R2,R1
     220 001362.  012702.  013012                    MOV.    #CNVT2S,R2.
     221 001366                                      CALL.   $CBTA.
     222.001372.  112720   000072                    MOVB.   #':,(R0)+               ;COLON.
     223 001376   010301                             MOV.    R3,R1                   ;CONVERT.SECONDS.TO.ASCII.
     224 001400   012702.  011012                    MOV.    #CNVT2,R2.
     225 001404                                      CALL.   $CBTA.
     226                                  ;
     227                                  ; OUTPUT.THE.5.DATA.LINES.
     228.                                 ;
     229 001410   012702.  000456'                   MOV.    #OUTBL,R2.              ;LINE.ADDRESS.AND.LENGTH.
     230 001414   012200                   OUTLOP:  MOV.    (R2)+,R0                ;ADDRESS.
     231 001416   001404                             BEQ.    1$                      ;DONE.IF.ZERO.
     232.001420   012201                             MOV.    (R2)+,R1                ;LENGTH.
     233 001422.                                     CALL.   TIOUT.                  ;OUTPUT.
     234 001426   000772.                            BR      OUTLOP.
     235                                  ;
     236 001430   000167   177130        1$:      JMP.    START.                  ;DONE.
     237                                  ;
     238                                  ;
     239                                  ; OUTPUT.A.STATISTICS.LINE.
     240                                  ;
     241 001434                           TIOUT:  QIOW$S.  #IO.WLB,#TILUN,#TIEF,,#IOSTAT,,<R0,R1,#40>
     242.001504                                      RTN.
     243                                  ;
```

```
                                        ;
                                        ; STATISTICS TYPE STRINGS
                                        ;
000504      103     125     122 CHS:    .ASCII  /CURRENT HOUR/
000507      122     105     116
000512      124     040     110
000515      117     125     122
000520      040     040     040 LHS:    .ASCII  /   LAST HOUR/
000523      114     101     123
000526      124     040     110
000531      117     125     122
000534      040     103     125 CDS:    .ASCII  / CURRENT DAY/
000537      122     122     105
000542      116     124     040
000545      104     101     131
000550      040     040     040 LDS:    .ASCII  /    LAST DAY/
000553      040     114     101
000556      123     124     040
000561      104     101     131
                                        ;
```

```
 88 000564                                  START:
 89 000564                                          RCVX$C  ,RCVDAT                 ;GET PACKET
 90                                         ; PARSE TYPE OF STATS - SET UP POINTER TO PROPER STATS AREA
 91 000572  012705  000000G                        MOV     #CHSTAT,R5              ;R5=STATS ADDRESS
 92 000576  012703  000504'                        MOV     #CHS,R3                 ;R3=TYPE OF STATS STRING
 93 000602  022767  000002  177176                 CMP     #2,RCVCHR
 94 000610  001040                                 BNE     MVTSTG                  ;CURRENT HOUR IF NO PARAM
 95 000612  022767  044103  177170                 CMP     #"CH,RCVPRM             ;BRANCH IF CURRENT HOUR
 96 000620  001434                                 BEQ     MVTSTG
 97 000622  012705  000000G                        MOV     #LHSTAT,R5              ;BRANCH IF LAST HOUR
 98 000626  012703  000520'                        MOV     #LHS,R3
 99 000632  022767  044114  177150                 CMP     #"LH,RCVPRM
100 000640  001424                                 BEQ     MVTSTG
101 000642  012705  000000G                        MOV     #CDSTAT,R5              ;BRANCH IF CURRENT DAY
102 000646  012703  000534'                        MOV     #CDS,R3
103 000652  022767  042103  177130                 CMP     #"CD,RCVPRM
104 000660  001414                                 BEQ     MVTSTG
105 000662  012705  000000G                        MOV     #LDSTAT,R5              ;BRANCH IF LAST DAY
106 000666  012703  000550'                        MOV     #LDS,R3
107 000672  022767  042114  177110                 CMP     #"LD,RCVPRM
108 000700  001404                                 BEQ     MVTSTG
109 000702  012705  000000G                        MOV     #CHSTAT,R5              ;CURRRENT HOUR IF
110 000706  012703  000504'                        MOV     #CHS,R3                 ;  BAD PARAMETER
111                                        ;
112                                        ; R5=ADDRESS OF STATS TABLE
113                                        ;
114                                        ; LINE 1
115                                        ;
116 000712  012704  000072'                MVTSTG: MOV     #TST,R4                 ;MOVE STATS TYPE TO LINE 1
117 000716  012702  000014                         MOV     #12.,R2
118 000722  112324                         1$:     MOVB    (R3)+,(R4)+
119 000724  077202                                 SOB     R2,1$
120                                        ; NUMBER OF BATCHES
121 000726  012700  000056'                        MOV     #NBT,R0
122 000732  016501  000000                         MOV     ST,BTC(R5),R1
123 000736  012702  027012                         MOV     #CNVTS,R2
124 000742                                         CALL    $CBTA
125                                        ;
126                                        ; SEE IF ANY STATS FOR THIS TIME PERIOD
127 000746  005765  000000                         TST     ST,BTC(R5)              ;CONTINUE IF THERE
128 000752  001030                                 BNE     DOLIN5
129 000754  022767  042103  177026                 CMP     #"CD,RCVPRM             ;CURRENT DAY?
130 000762  001003                                 BNE     2$                      ;NO
131 000764  012705  000000G                        MOV     #CHSTAT,R5              ;YES - TRY CURRENT HOUR
132 000770  000406                                 BR      3$
133 000772  022767  042114  177010         2$:     CMP     #"LD,RCVPRM             ;LAST DAY?
134 001000  001005                                 BNE     4$                      ;NO
135 001002  012705  000000G                        MOV     #LHSTAT,R5              ;YES - TRY LAST HOUR
136 001006  005765  000000                 3$:     TST     ST,BTC(R5)              ;CONTINUE IF NEW TIME
137 001012  001010                                 BNE     DOLIN5                  ;  PERIOD HAS STATS
138 001014  012700  000046'                4$:     MOV     #LINE1,R0               ;ELSE JUST OUTPUT
139 001020  012701  000040                         MOV     #SL1,R1                 ;  FIRST LINE
140 001024                                         CALL    TIOUT
141 001030  000167  177530                         JMP     START                   ;AND FINISH
142                                        ;
```